

Presentacions col·laboratives amb l'eyeOS

Projecte de final de carrera
Enginyeria Tècnica en Informàtica de Gestió
Eduard Gamonal Capdevila
16 de setembre de 2011

Presentacions col·laboratives amb l'eyeOS

Eduard Gamonal Capdevila

Director Gerard Muñoz (eyeOS)

Ponent Carles Farré (ESSI, UPC)

Projecte de final de carrera - Enginyeria Tècnica en Informàtica de Gestió

Facultat d'Informàtica de Barcelona - Universitat Politècnica de Catalunya

Desenvolupat a eyeOS (La Rambla, 140, 4C, 08002 Barcelona, Catalunya)

Copyright

Aquesta obra està subjecta a una llicència [Reconeixement-CompartirIgual 3.0 No adaptada de Creative Commons](http://creativecommons.org/licenses/by-sa/3.0/). Per veure una còpia d'aquesta llicència, visiteu <http://creativecommons.org/licenses/by-sa/3.0/> o envieu una carta a Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.

Totes les marques d'aquest document pertanyen als seus legítims propietaris.

Comentaris

Podeu enviar comentaris o suggeriments sobre aquest document a eduard.gamonal@est.fib.upc.edu

Data de publicació i versió del programari

Publicat el 16 de setembre de 2011.

eyeMeeting 1.0 per l'eyeOS 2.5

Agraïments

Voldria agrair l'ajuda rebuda durant les darreres setmanes que ha fet possible el projecte:

A les diverses persones que han revisat aquest document amb cura i detall.

Als companys de l'oficina de l'eyeOS per la confiança posada en mi i l'orientació inicial. Especialment en Gerard Muñoz, el director del projecte; en Jose Carlos Norte, el CTO; l'equip de disseny i directiu i la Mònica Casanovas.

A la comunitat de Qooxdoo per la seva rebuda i explicacions precises de la biblioteca gràfica.

A en Marc Puts, Universitat de Maastricht, pel seu contrib SVG i les seves idees.

Al professor Ernest Teniente per l'orientació en les primeres passes amb el projecte i l'entusiasme transmès per l'enginyeria del programari.

Al meu tutor del projecte i professor Carles Farré, per la seva flexibilitat i orientació.

Índex de continguts

1	Introducció.....	1
1.1	Què és l'eyeOS	1
1.2	Objectius	2
1.2.1	Generals	2
1.2.2	Específics	3
1.3	Antecedents	3
1.4	Organització de la memòria	3
2	Planificació.....	5
2.1	Fases	5
2.1.1	Fase d'inici	5
2.1.2	Fase de planificació	5
2.1.3	Fase d'execució	5
2.1.4	Fase de tancament	6
2.2	Estimació de les hores	6
2.3	Anàlisi de riscos	9
3	Anàlisi de requisits.....	11
3.1	Parts interessades	11
3.2	Abast	12
3.3	Requeriments funcionals	12
3.3.1	Accions sobre la presentació	12
3.3.2	Accions sobre els participants	13
3.3.3	Accions sobre una diapositiva	13
3.4	Requeriments no funcionals	13
4	Especificació.....	15
4.1	Model Conceptual	15
4.2	Actors	17
4.3	Casos d'ús	17
4.3.1	Gestionar la presentació	18
4.3.2	Gestionar els participants	23
4.3.3	Gestionar pintades	25
4.4	Contractes	26
4.5	Diagrames d'estat	33
5	Disseny.....	35
5.1	L'entorn: eyeOS	35
5.1.1	Vista estàtica	35
5.1.1.1	Introducció	35
5.1.1.2	L'eyeOS és una aplicació web	36
5.1.1.3	L'eyeOS i les seves aplicacions són distribuïts	37
5.1.1.4	Arquitectura de l'eyeOS	37
5.1.2	Vista dinàmica	39
5.1.2.1	Visió general	39
5.1.2.2	Crida local	39
5.1.2.3	Crida remota	40
5.2	L'aplicació del projecte	41
5.2.1	Arquitectura física	41

5.2.2 Arquitectura lògica: disseny en capes	46
5.2.2.1 Tecnologies utilitzades en cada capa	46
5.2.2.2 Disseny de la capa de presentació web	48
Disseny extern	49
Disseny intern	58
5.2.2.3 Disseny de la capa de domini	79
5.2.2.4 Disseny de la capa d'accés a dades	94
6 Implementació.....	99
6.1 Entorn de desenvolupament	99
6.1.1 Instal·lació	99
6.1.2 Desenvolupament	99
6.1.3 Depuració	100
6.2 Exemples d'ús de les tecnologies utilitzades	102
6.2.1 Creació de la finestra i dels seus components	103
6.2.2 Suport per documents d'oficina	104
6.2.3 SVG	106
6.2.4 Crida remota	110
6.2.5 Listeners	110
6.2.6 NetSync i d-bus	111
6.2.7 Compartició de pintades	112
6.2.8 Servei de fitxers JavaScript al client	113
6.2.9 Connexió amb la base de dades	114
7 Proves del sistema.....	115
7.1 Disseny dels jocs de proves i resultats	115
7.1.1 Proves funcionals	115
7.1.2 Proves de rendiment	118
8 Instal·lació i execució del sistema.....	121
8.1 Requeriments	121
8.2 Procediment	121
9 Manual de l'usuari.....	123
9.1 Quant el manual	123
9.2 Introducció	124
9.2.1 Importar una presentació	124
9.2.2 Gestionar els participants	128
9.2.3 Fer pintades	128
9.3 Autor	129
9.4 Llicència	129
10 Planificació real i valoració econòmica.....	131
10.1 Anàlisi de costos	133
10.1.1 Recursos humans	133
10.1.2 Recursos materials	134
10.1.3 Cost total	135
11 Conclusions i treball futur.....	137
12 Bibliografia i referències.....	141
12.1 Obres consultades	141
12.2 Recursos electrònics	142

Índex d'il·lustracions

Il·lustració 1: Visió general de l'eyeOS.....	1
Il·lustració 2: Diagrama de Gantt del projecte.....	8
Il·lustració 3: Model conceptual.....	15
Il·lustració 4: Jerarquia d'actors.....	17
Il·lustració 5: Casos d'ús de gestionar presentació.....	18
Il·lustració 6: Diagrama de seqüència de sistema d'iniciar presentació.....	19
Il·lustració 7: Diagrama de seqüència del sistema per crear una presentació.....	20
Il·lustració 8: Diagrama de seqüència del sistema per obtenir una presentació.....	20
Il·lustració 9: Diagrama de seqüència del sistema per actualitzar una presentació.....	21
Il·lustració 10: Diagrama de seqüència del sistema per esborrar una presentació.....	21
Il·lustració 11: Diagrama de seqüència del sistema per mostrar una diapositiva.....	22
Il·lustració 12: Casos d'ús de gestionar participants.....	23
Il·lustració 13: Diagrama de seqüència del sistema per crear un participant.....	24
Il·lustració 14: Casos d'ús de gestionar pintades.....	25
Il·lustració 15: Diagrama d'estats dels tipus d'usuari.....	33
Il·lustració 16: Diagrama d'estats d'una presentació.....	34
Il·lustració 17: Distribució de capes en els nodes.....	38
Il·lustració 18: Crida local.....	39
Il·lustració 19: Crida remota.....	40
Il·lustració 20: Dues crides consecutives amb respostes diferents.....	41
Il·lustració 21: Dues crides agrupades.....	41
Il·lustració 22: Vista de desplegament de l'eyeMeeting.....	43
Il·lustració 23: Wireframe de la finestra principal (vista d'orador).....	50
Il·lustració 24: Wireframe de la pantalla principal (vista d'oient).....	51
Il·lustració 25: Wireframe de la finestra de configuració.....	51
Il·lustració 26: Wireframe del navegador de fitxers.....	52
Il·lustració 27: Mapa navegacional d'àmbit de sistema.....	52
Il·lustració 28: Mapa navegacional de gestionar presentació (1).....	53
Il·lustració 29: Mapa navegacional de gestionar presentació (2).....	54
Il·lustració 30: Mapa navegacional de gestionar presentació (3).....	54
Il·lustració 31: Mapa navegacional de gestionar presentació (4).....	55
Il·lustració 32: Mapa navegacional de gestionar presentació (5).....	55
Il·lustració 33: Mapa navegacional de gestionar presentació (6).....	56
Il·lustració 34: Mapa navegacional de gestionar participants.....	57
Il·lustració 35: Mapa navegacional de gestionar pintades.....	58
Il·lustració 36: Diagrama de classes de presentació (1).....	61
Il·lustració 37: Diagrama de classes de presentació (2).....	61
Il·lustració 38: Diagrama de classes de presentació (3).....	62
Il·lustració 39: Diagrama de seqüència: onImport.....	64
Il·lustració 40: Diagrama de seqüència: onPresentationImported.....	64
Il·lustració 41: Diagrama de seqüència onNetSyncMessageReceived (1).....	65
Il·lustració 42: Diagrama de seqüència onNetSyncMessageReceived (2, continuació).....	66
Il·lustració 43: Diagrama de seqüència de showSlide (1A).....	67
Il·lustració 44: Diagrama de seqüència de showSlide (1B).....	68
Il·lustració 45: Diagrama de seqüència de showSlide (2).....	69
Il·lustració 46: Diagrama de seqüència de showSlide (3) -- create-slide-svg.....	70

Il·lustració 47: Diagrama de seqüència de broadcast.....	71
Il·lustració 48: Diagrama de seqüència de startDraw.....	72
Il·lustració 49: Diagrama de seqüència de draw.....	73
Il·lustració 50: Diagrama de seqüència d' endDraw.....	73
Il·lustració 51: Diagrama de seqüència d' addDoodle.....	74
Il·lustració 52: Diagrama de seqüència de createSourceList.....	75
Il·lustració 53: Diagrama de seqüència d' onParticipantsReceived.....	75
Il·lustració 54: Diagrama de seqüència de createItem.....	76
Il·lustració 55: Diagrama de seqüència de bindItem.....	77
Il·lustració 56: Diagrama de seqüència de onWorkGroupsListReceived.....	78
Il·lustració 57: Diagrama de classes.....	83
Il·lustració 58: Diagrama de seqüència d' importPresentation.....	85
Il·lustració 59: Diagrama de seqüència de nextSlide.....	86
Il·lustració 60: Diagrama de seqüència de getSlide.....	86
Il·lustració 61: Diagrama de seqüència d' initializeAll.....	87
Il·lustració 62: Diagrama de seqüència de getSlideImageBase64.....	88
Il·lustració 63: Diagrama de seqüència de sharePresentation.....	89
Il·lustració 64: Diagrama de seqüència d'initPresentation.....	90
Il·lustració 65: Diagrama de seqüència d' updatePresentation (1).....	92
Il·lustració 66: Diagrama de seqüència d' updatePresentation (2).....	93
Il·lustració 67: Diagrama de seqüència de sendInvitations.....	94
Il·lustració 68: Comparació entre dissenys ideal i necessari de la capa d'abstracció de dades.....	95
Il·lustració 69: Consola del Firebug.....	100
Il·lustració 70: Vista de les dades POST en la consola del Firebug.....	101
Il·lustració 71: Pestanya d'script i punt d'interrupció en el Firebug.....	101
Il·lustració 72: Finestra principal de l'eyeMeeting.....	104
Il·lustració 73: Finestra inicial.....	125
Il·lustració 74: Navegador de fitxers.....	125
Il·lustració 75: Finestra de l'orador.....	126
Il·lustració 76: Finestra de l'oient.....	127
Il·lustració 77: Configuració.....	127
Il·lustració 78: Gestió dels participants.....	128

Índex de taules

Taula 1: Iteracions en la fase d'execució.....	6
Taula 2: Estratègies de tractament de la jerarquia.....	96
Taula 3: Recursos i salaris.....	133
Taula 4: Descomposició de tasques i costos.....	134

Índex de fragments de codi

Codi 1: SQL de l'eyeMeeting.....	98
Codi 2: Canvis en la crida al sistema de l'OpenOffice.....	99

Codi 3: Finestra principal.....	103
Codi 4: Callback en fer clic al botó d'importació.....	104
Codi 5: Crida remota per convertir un document d'oficina.....	105
Codi 6: Conversió de documents d'oficina -- costat del servidor.....	105
Codi 7: Sortida del terminal amb la llista de fitxers generats per l'OOo.....	106
Codi 8: Estructura del node SVG.....	107
Codi 9: Creació del component svg.embed.Svg.....	108
Codi 10: Creació dels components gràfics de les diapositives (1).....	108
Codi 11: Creació dels components gràfics de les diapositives (2).....	109
Codi 12: Crida remota.....	110
Codi 13: Listener.....	110
Codi 14: Funció de callback anònima.....	111
Codi 15: Enviament de missatge amb NetSync.....	111
Codi 16: Alta al canal de NetSync.....	112
Codi 17: Baixa del canal de NetSync.....	112
Codi 18: Acabament de la pintada.....	112
Codi 19: Enviament de la pintada.....	113
Codi 20: En rebre un missatge.....	113
Codi 21: Afegir una pintada.....	113
Codi 22: Servei de fitxers JavaScript al client.....	114
Codi 23: Connexió amb la base de dades.....	114

1 INTRODUCCIÓ

L'eyeOS és un escriptori web lliure que segueix el concepte de *cloud computing*¹. El seu potencial rau en permetre el treball col·laboratiu i concurrent entre usuaris, la disponibilitat de les dades que allotja i la portabilitat, entre d'altres.

En aquest context s'ha desenvolupat una aplicació dins un navegador per fer presentacions en reunions de manera col·laborativa i a distància.

1.1 Què és l'eyeOS²

Per l'usuari final, l'eyeOS és un escriptori web amb una interfície semblant a la d'un sistema operatiu popular d'un ordinador. Tot s'emmagatzema en el núvol de manera que finalment l'usuari no depengui del programari específic de l'ordinador. Un navegador avançat hauria de ser suficient.



Il·lustració 1: Visió general de l'eyeOS

És un sistema útil per intranets, educació, en qualsevol entorn que requereixi treball en equip o per usuaris independents que necessitin tenir els seus documents i aplicacions disponibles arreu.

Per a un desenvolupador, l'eyeOS és una aplicació rica a internet que ofereix un

¹ Bibliografia i referències[Youseff]

² Bibliografia i referències [Gamonal]

conjunt d'eines per desenvolupar noves aplicacions web de manera semblant a com s'escriu el programari d'escriptori.

Les aplicacions s'escriuen en JavaScript i PHP i són distribuïdes: part del codi va al client, part al servidor. L'eyeOS utilitza Qooxdoo³ en comptes de simple JavaScript perquè ofereix orientació a objectes tradicional i permet escriure una interfície gràfica com si es tractés de GTK o Swing.

1.2 Objectius

Sovint les organitzacions tenen un alt grau de dispersió geogràfica i per tant fins i tot també temporal entre els seus membres. És molt habitual fer presentacions per informar massivament però sovint han de ser presencials i obliguen els assistents a desplaçar-se o a renunciar a la presentació per qüestions logístiques. Les organitzacions que a dia d'avui utilitzen l'eyeOS acostumen a tenir aquest problema i necessiten fer reunions i presentacions virtuals. Per exemple, les Biblioteques de Barcelona o alguns departaments d'Orange i IBM no disposen d'una eina que els ho permeti i han de recórrer a programari extern a la seva plataforma. Perjudica la interoperabilitat del seu sistema i provoca una fragmentació i un cost afegits completament indesitjables.

Amb aquest projecte es desenvoluparà una aplicació que permeti als usuaris de l'eyeOS poder fer una reunió en línia amb una presentació. Un orador organitzarà les invitacions i executarà la presentació. Els assistents podran, si l'orador ho permet, pintar sobre les diapositives. El canal de comunicació principal recomanat és la veu, per telèfon tradicional o VoIP, però l'aplicació no ho ha d'implementar.

1.2.1 Generals

S'ha de dotar l'eyeOS d'una aplicació per fer reunions a distància.

- Ha d'incrementar la interacció entre orador i oients.
- Ha d'explotar el treball col·laboratiu.
- Ha d'estar integrada en l'eyeOS (coherència terminològica, aspecte i comportament, etc).

³Qooxdoo (pàg 47)

1.2.2 Específics

El projecte s'ha desenvolupat de manera iterativa amb 3 objectius concrets:

- Sistema base que importi un fitxer ODP o PPT(X) i que permeti navegar per les diapositives.
- Sistema de gestió d'usuaris de l'aplicació integrat en l'eyeOS.
 - Selecció de contactes convidats.
 - Compartició del fitxer.
 - Enviament d'invitacions.
- Sistema de compartició de pintades sobre les diapositives.

Les decisions que s'han pres s'han basat en el compliment d'aquests objectius, que porten al compliment de l'objectiu global.

1.3 Antecedents

Existeixen algunes aplicacions i conceptes similars a l'eyeMeeting. El més rellevant és *webinar*, un sistema de conferència remota en línia que pot ser implementat amb un sistema basat en web. Normalment és unidireccional però existeixen implementacions en forma d'aplicació col·laborativa. L'Adobe Connect i el Tokbox són probablement les més conegudes. Els usuaris poden recórrer també a l'Skype, sistemes de compartició de l'escriptori, etc. L'openMeeting, The Big Blue Button i el WebEX de Cisco són també molt populars.

En l'eyeOS no existeix cap aplicació per fer presentacions de manera col·laborativa. Aquesta plataforma està creixent en les intranets d'organitzacions de tot tipus on, habitualment, cal fer presentacions. Això suposa una mancança important que aquest projecte soluciona.

1.4 Organització de la memòria

Tot i que el projecte s'ha fet amb iteracions, la memòria presenta el resultat final en les fases del cicle de vida tradicional del programari per facilitar-ne la lectura. Al principi de cada secció s'ofereix una visió general per posar el lector en context.

Convencions

- *Itàlica*

Indica nous termes, URLs, noms de fitxers, extensions, noms de classes o termes per definir.

- **Amplada constant**

Indica codi en el seu significat més ample. Inclou ordres, opcions, variables, funcions...

- **Amplada constant i negreta**

Indica ordres o text que el lector s'hauria d'escriure literalment.

Exemples de codi

Tots els exemples de codi són lliures i no requereixen cap permís de l'autor. S'agraeix, però no es requereix, atribució. Es pot incloure el títol, autor i universitat en què s'ha fet aquesta memòria:

Presentacions col·laboratives amb l'eyeOS, d'Eduard Gamonal Capdevila. Facultat d'Informàtica de Barcelona. Universitat Politècnica de Catalunya (Barcelonatech) i eyeOS. Setembre 2011.

2 PLANIFICACIÓ

En aquest capítol es descriu quina planificació temporal s'ha seguit pel projecte i un anàlisi de riscos. La valoració econòmica i la planificació real són al capítol 10.

2.1 Fases

La gestió del projecte s'ha dividit en unes fases típiques d'inici, planificació, execució i tancament.

2.1.1 Fase d'inici

Definició del producte 19 de Novembre – 11 de febrer

Primeres converses amb l'empresa per determinar el tema del projecte, fer un anàlisi de requisits inicial i pactar la data d'entrada a l'oficina.

Formació 9 març – 18 de març

Trobar els manuals, entendre l'eyeOS, determinar llocs on trobar ajuda i fer una aplicació de prova amb una interfície gràfica senzilla però no elemental i que interactuï amb la base de dades.

2.1.2 Fase de planificació

Planificació 11 de febrer – 9 març

Determinar la planificació en detall i refinar els requisits. Ha començat abans d'acabar la fase d'inici per accelerar el projecte. No obstant, no interfereix amb cap activitat.

2.1.3 Fase d'execució

21 març – 14 juliol

Web Engineering is the application of systematic and quantifiable approaches (concepts, methods, techniques, tools) to cost effective requirements analysis, design, implementation, testing, operation, and maintenance of high quality Web applications. [Kappel]

Kappel et també diu que un procés d'enginyeria web hauria de complir les

següents característiques:

- Ser incremental
- Tenir canvis freqüents
- Ocupar un interval curt de temps.

El desenvolupament de l'aplicació encaixa en aquesta plantilla i, per tant, s'ha fet de manera iterativa incremental. Hi ha hagut 3 iteracions, que coincideixen amb els objectius específics, o 4 si es prefereix incloure les activitats de la fase de planificació com una iteració més.

Iteració	Artefactes
Especificació, disseny, implementació i proves del sistema base	Documentació i codi funcional
Especificació, disseny, implementació i proves del sistema de gestió d'usuaris	Documentació i codi funcional
Especificació, disseny, implementació i proves del sistema de compartició de pintades sobre les diapositives.	Documentació i codi funcional

Taula 1: Iteracions en la fase d'execució

2.1.4 Fase de tancament

11 juliol – 20 agost Escriure el document final de la memòria.

30 juliol Presentació a eyeOS de la nova aplicació.

20 – 24 agost Revisió de la memòria.

16 de setembre Defensa.

2.2 Estimació de les hores

Amb les activitats fixades, les hores aproximades dedicades al projecte són 505 repartides de la següent manera

- Fase d'inici

Definició del producte	15h
Formació	50h

- Fase de planificació

Planificació	15h
--------------	-----

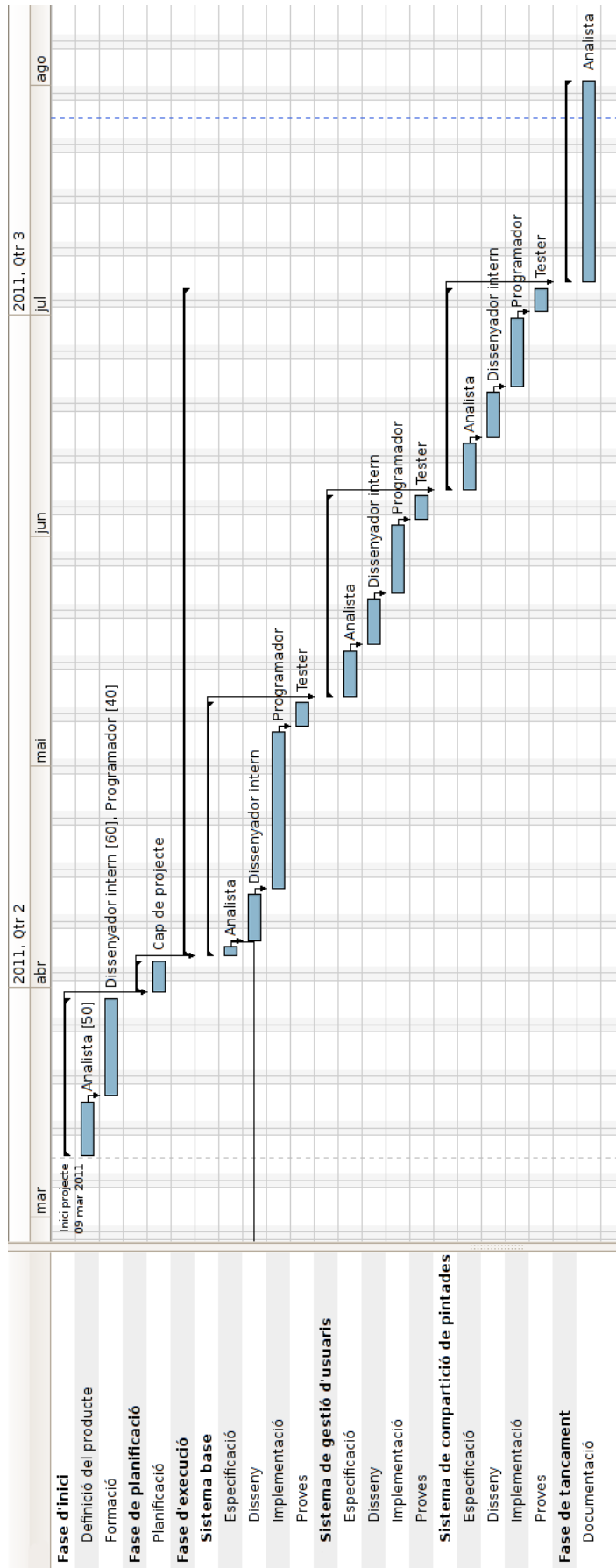
- Fase d'execució

Especificació, disseny, implementació i proves del sistema base	125h (5 setmanes)
Especificació, disseny, implementació i proves del sistema de gestió d'usuaris	100h
Especificació, disseny, implementació i proves del sistema de compartició de pintades sobre les diapositives.	100h

- Fase de tancament

Memòria i preparació defensa	100h
------------------------------	------

Les activitats en detall es poden veure en la Il·lustració 2: Diagrama de Gantt del projecte



Il·lustració 2: Diagrama de Gantt del projecte

2.3 Anàlisi de riscos

Tots els projectes tenen riscos, coses que poden no sortir bé, però sense riscos no es pot avançar⁴. Amb una previsió correcta es poden gestionar d'una manera més segura i fer els canvis de manera controlada.

Previsió temporal inadequada

És possible que durant la fase d'execució no es compleixin els terminis. L'eyeOS és un sistema completament nou pel desenvolupador i d'una naturalesa diferent a l'estudiada en profunditat a la carrera.

Probabilitat: molt alta.

Impacte: menor si es detecta a temps; crític altrament.

Solució: Assignar-hi més recursos si n'hi ha i limitar l'abast del projecte. Tornar a estimar les hores.

Canvis inesperats en la definició del producte

És possible que l'aplicació no estigui completament definida abans de començar amb la implementació i que el CTO, el director del projecte o l'equip de disseny extern demanin fer alguns canvis.

Probabilitat: alta.

Impacte: mitjà.

Solució: comprovar que es pot canviar l'abast sense una desviació important de la planificació, proposar una reunió per definir l'abast de manera completa i acceptar els canvis necessaris.

No es pot instal·lar el suport per oficina

És possible que no s'aconsegueixi instal·lar el suport per oficina en l'ordinador de treball habitual.

Probabilitat: baixa.

Impacte: crític.

Solució: virtualització.

4 [ER] Volere

No es pot trobar una manera de pintar sobre les diapositives

És possible que no existeixi cap manera d'implementar les pintades sobre les diapositives en l'eyeOS amb els recursos previstos.

Probabilitat: mitjana

Impacte: crític

Solució: tornar a l'anàlisi de requeriments i buscar una altra manera de complir amb l'objectiu. És possible que no calgui pintar a mà alçada sinó que es puguin crear formes o fer que el cursor destaquï a petició de l'usuari, acció que sí que és factible.

Interrupció del desenvolupament per causes externes

Per malaltia, operacions en l'oficina o servidors, problemes tècnics...

Probabilitat: baixa.

Impacte: baix.

Solució: gestionar la planificació temporal. Si la interrupció és de més d'una setmana i no es poden recuperar les hores, comprovar la limitació de l'abast

Qualitat insuficient

El producte final no compleix els requeriments. Aquest risc es pot detectar al final de cada iteració.

Probabilitat: molt baixa

Impacte: crític

Solució: tornar a l'especificació i al disseny de la iteració afectada. Comprovar que no hi haurà desviacions importants en la planificació.

3 ANÀLISI DE REQUISITS

Aquest capítol conté una visió general del projecte, informació detallada sobre les parts interessades i la descripció de les característiques principals. Està centrat en els objectius i és la referència bàsica que regeix la presa de decisions durant tot el procés de desenvolupament. Ja hi ha reflectits tots els canvis en l'abast acordats durant el desenvolupament.

3.1 Parts interessades

- *Orador*. Un usuari de l'eyeOS interessat en fer una presentació i que disposa de material organitzat per ensenyar. Espera un rol més important que els oients i poder fer una presentació en línia dins la seva plataforma.
- *Oient*. Un usuari de l'eyeOS que assisteix a la presentació. Espera una experiència d'usuari similar o millor a la que tindria amb les aplicacions del capítol. S'ha produït un error: No s'ha trobat la font de referència. S'ha produït un error: No s'ha trobat la font de referència.
- *Facultat d'Informàtica de Barcelona, UPC*. Espera el nivell de qualitat que caracteritza la facultat en una feina de 450 hores, que inclouen la documentació, per realitzar un sistema d'informació únic i amb futur.
- *eyeOS*. Espera un programa de qualitat, per la seva plataforma, ben planificat i amb un pressupost tancat, que solucioni el problema plantejat. Espera incrementar el nombre d'usuaris del seu sistema lliure amb una aplicació atractiva i útil. Aquest projecte s'ha dut a terme dins l'empresa. En detall, els interessats són:
 - *Gestor de projectes* – tutor. Espera que es compleixi el temps, fites i qualitat i que els canvis es gestionin adequadament.
 - *CTO* – espera una aplicació que faci un ús correcte de l'entorn i que compleixi els criteris de qualitat de l'empresa.
 - *Equip de disseny* – espera que la interfície gràfica segueixi les directrius de l'eyeOS tant en aspecte com en usabilitat.

- *Equip de direcció* – espera una aplicació que aportí nou valor a l'eyeOS.

3.2 Abast

L'aplicació ha de permetre a un orador posar en marxa una presentació de manera que un grup d'assistents escollits pugui veure-la en línia i interactuar-hi. Ha de funcionar dins l'eyeOS, en general en un navegador. No ha de proporcionar mitjans per transmetre veu ni vídeo. Els participants poden compartir algunes dades segons els permisos que l'orador doni.

El navegador de referència ha de ser multiplataforma, lliure, ha de respectar els estàndards oberts i ha de promoure el web obert per sintonia amb la plataforma de treball, qualitat i garanties de futur. S'utilitzarà Mozilla Firefox 4 o superior.

3.3 Requeriments funcionals

3.3.1 Accions sobre la presentació

1. Gestió de la presentació
 1. Ha de permetre importar una presentació en format d'OpenOffice o de Microsoft Office.
 2. Ha de permetre al propietari del fitxer configurar-la.
 1. Editar-ne el títol.
 2. Editar-ne el dia i l'hora en què es farà.
2. Ús de la presentació
 1. Ha de permetre navegar endavant i endarrere.
 2. Ha de permetre a l'orador començar i acabar la presentació.
 1. Quan la presentació està iniciada, l'orador controla quina diapositiva veu tothom.
 2. Quan no està iniciada, o està acabada, el participant⁵ pot navegar per les diapositives.
 3. Ha de permetre a l'orador controlar quina diapositiva veuen els participants.
 4. Ha de permetre a l'orador veure quins continguts té la diapositiva següent.
 5. Ha de permetre a l'orador veure quins continguts té la diapositiva

⁵ Vegeu Il·lustració 4: Jerarquia d'actors

anterior.

3.3.2 Accions sobre els participants

1. Ha de permetre afegir un participant
2. Ha de permetre esborrar un participant
3. Ha de permetre canviar el tipus d'un participant.
 1. Un participant pot ser orador o oient.
 2. Un orador pot gestionar la presentació.
 3. Un oient pot ser actiu o passiu. Els actius poden enviar informació a tothom. Els passius només poden veure la presentació i la informació que es comparteixi.
4. Ha de permetre l'enviament d'invitacions i notificacions de canvis en la configuració de la presentació als participants.

3.3.3 Accions sobre una diapositiva

1. Ha de permetre a un usuari fer pintades a mà alçada sobre la diapositiva que estigui a la vista.

3.4 Requeriments no funcionals

1. L'aplicació ha de funcionar en un navegador que pugui executar l'eyeOS i que permeti renderitzar SVG. Mozilla Firefox 4 o superior.
2. L'aplicació ha de ser extensible perquè els desenvolupadors de l'eyeOS o la comunitat puguin afegir-hi funcions.
3. La interfície gràfica de l'aplicació ha de seguir les directrius de l'equip de disseny.
 - En usabilitat
 - En aspecte
4. S'ha d'acabar abans de l'11 de juliol, amb el contracte del desenvolupador.
5. No hi ha requisits de rendiment perquè els elements que el determinen són aliens a l'aplicació:
 - Velocitat de resposta del convertidor de l'OpenOffice.
 - Ample de banda i latència de la xarxa.No obstant, s'ha de construir l'aplicació de manera que es minimitzi l'ús dels elements que poden fer baixar el rendiment.
6. L'aplicació s'ha de lliurar amb:
 - Manual d'instal·lació, destinat a l'administrador de sistemes, per la

manera en què s'instal·len les aplicacions en l'eyeOS 2.5

- Manual d'usuari, breu i que expliqui totes les funcions de l'aplicació.

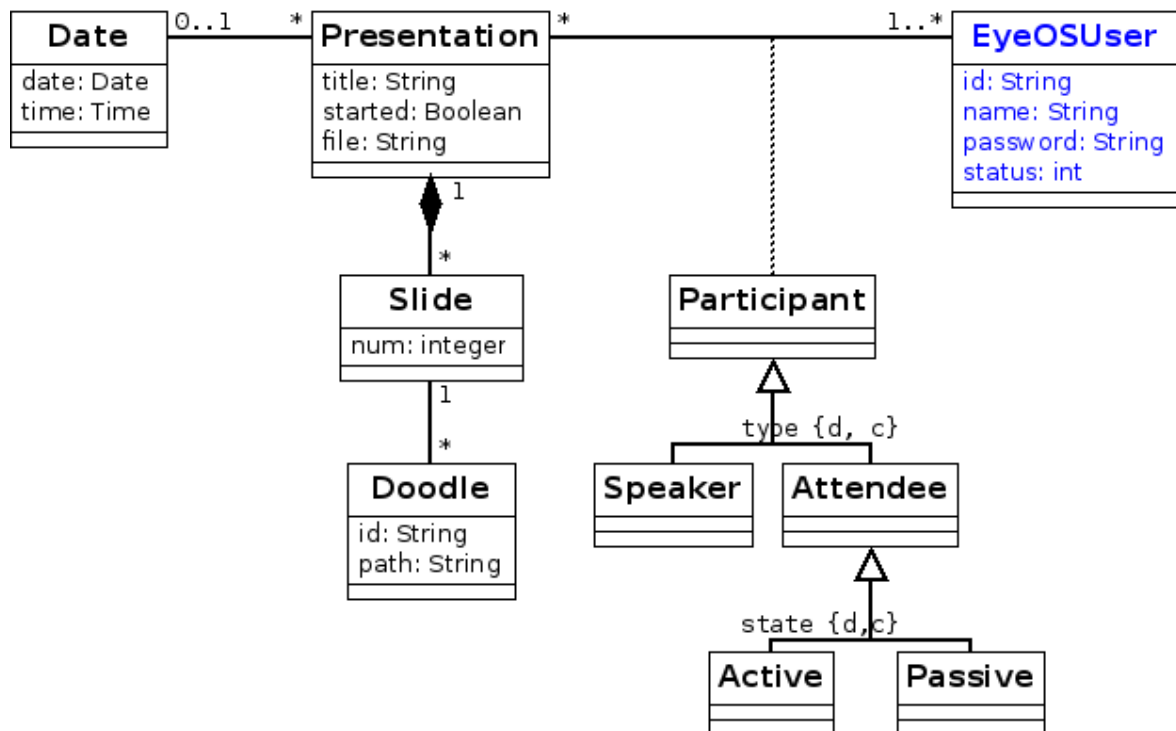
4 ESPECIFICACIÓ

4.1 Model Conceptual

En una primera aproximació al problema es va fer una especificació extensa i més complexa del necessari. En posteriors redefinicions de l'abast es va anar corregint fins arribar a un resultat més simple i adequat pel problema inicial.

La Il·lustració 3 és el model conceptual de l'aplicació. Conté una representació dels conceptes del món real, no dels conceptes del programa. L'objectiu és comprendre millor el problema.

EyeOSUser és en un color diferent perquè no forma part de l'aplicació però s'utilitza.



Il·lustració 3: Model conceptual

Restriccions d'integritat:

- No existeixen dues Presentation amb els mateixos Participant i Data

Clau externa

- Data: date, time
- Presentation: file

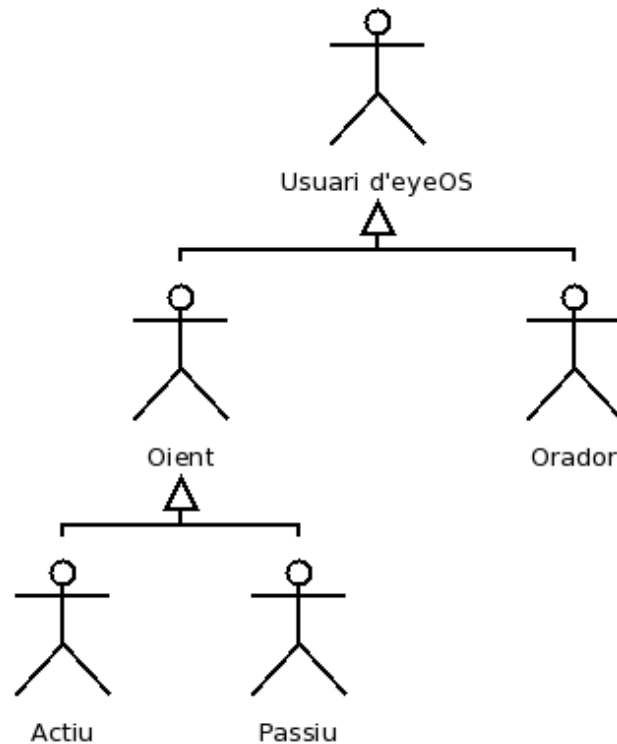
- Slide: num

Les altres classes utilitzen els identificadors dels objectes (oid) com clau externa.

- *Presentation*
Representa el material que s'utilitza en la reunió. Tot i el nom, cal adonar-se que no és necessàriament un fitxer de l'OpenOffice, sinó una entitat en l'especificació d'aquesta aplicació. La seva data és el moment en què l'orador desitja començar la presentació.
- *Slide*
Representa una diapositiva de la presentació. Per cadascuna se n'enregistra l'ordinal.
- *Doodle*
Representa una pintada en una diapositiva. Se n'enregistra un identificador i un camí en forma de coordenades.
- *Participant*
Representa un usuari de l'eyeOS en una presentació.
- *Date*
Representa una data i una hora.
- *Speaker*
Representa l'usuari amb permisos menys restrictius. En aquest document és sinònim d'«orador»
- *Attendee*
Representa els participants que no són Speaker. Les classes de la seva especialització corresponen a diferents nivells de permisos. En aquest document és sinònim d'«oient», que pot ser «actiu» o «passiu».
- *EyeOSUser*
Representa un usuari de l'eyeOS.

4.2 Actors

Els actors que interactuaran amb el sistema són l'usuari de l'eyeOS, l'orador, el participant actiu i el passiu. Cal no confondre'ls amb tipus d'usuari d'eyeOS. Un usuari d'eyeOS, d'acord amb el tipus que li sigui assignat, esdevé un actor o un altre.



Il·lustració 4: Jerarquia d'actors

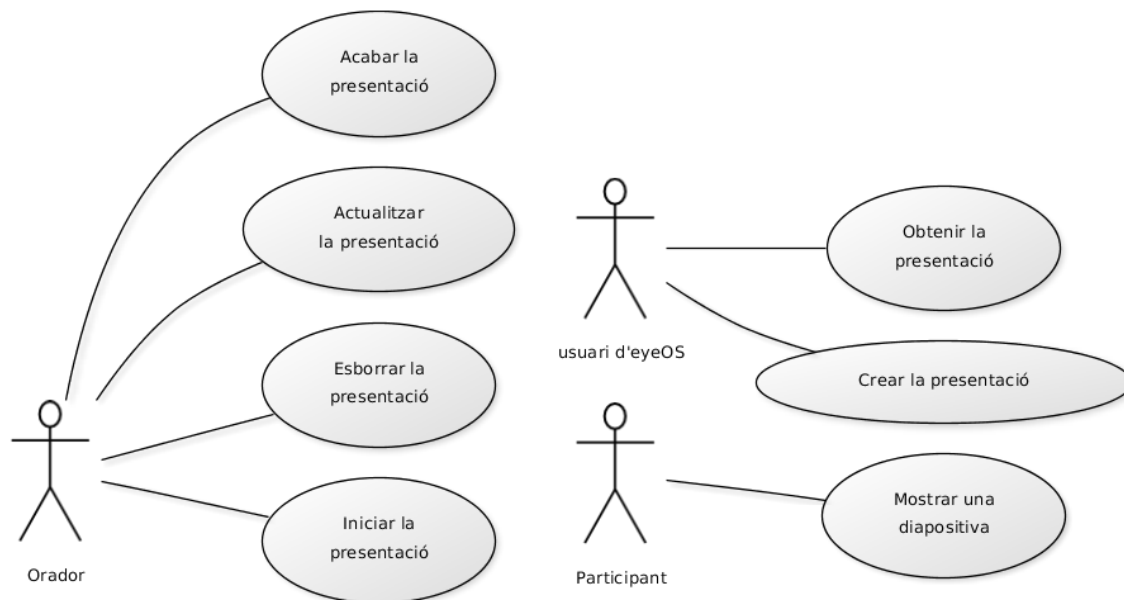
4.3 Casos d'ús

En aquest capítol s'especifica la interacció entre actors i el sistema. S'ha parat especial atenció a l'entrada i sortida de dades i a com el sistema ha de complir els requeriments.

En alguns casos la redacció pot resultar un tant redundant. S'ha hagut de fer així per consistència amb el model. Per exemple, conceptualment el sistema permet que un usuari participi en diverses presentacions simultànies i, per tant, oracions com «el participant escull una diapositiva» donen peu a ambigüitats. Per solucionar-ho en l'especificació, ha calgut concretar que l'usuari sempre escull una presentació. Durant el disseny s'han pres decisions per limitar l'aplicació i simplificar la solució d'aquestes ambigüitats.

Estan agrupats en 3 paquets que contenen les interaccions típiques CRUD⁶ i els casos d'ús particulars relacionats [Cockburn].

4.3.1 Gestionar la presentació



Il·lustració 5: Casos d'ús de gestionar presentació

Iniciar

Un orador pot iniciar una presentació. Quan una presentació està iniciada, l'Orador pot canviar les diapositives dels seus participants.

Actors: Orador

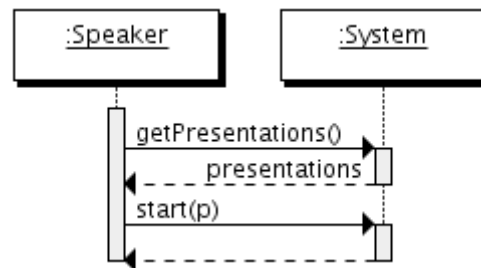
Activació: un orador vol iniciar la presentació.

Precondició: la presentació no està ja iniciada.

Escenari principal:

1. L'Orador indica al Sistema que s'ha d'iniciar una presentació
2. El sistema mostra les presentacions
3. L'Orador indica al sistema que iniciï la presentació p.
4. El Sistema enregistra el nou estat de la presentació

⁶ Create, Retrieve, Update, Delete



Il·lustració 6: Diagrama de seqüència de sistema d'iniciar presentació

Acabar

Un orador pot acabar una presentació. Quan una presentació està acabada, o no ha estat iniciada, l'orador no té control sobre les diapositives dels participants. Els participants ignoren ordres fins que la presentació estigui iniciada. Poden navegar lliurement per les diapositives en local.

Actors: Orador

Activació: Un orador vol acabar la presentació.

Precondició: la presentació està iniciada.

Escenari principal:

1. L'Orador indica al Sistema que s'ha d'acabar una presentació
2. El sistema mostra les presentacions
3. L'Orador indica al sistema que acabi la presentació p.
4. El Sistema enregistra el nou estat de la presentació.

Nota: el diagrama és similar a l'anterior.

Crear

Un usuari de l'eyeOS pot crear una presentació a partir d'un fitxer que l'OpenOffice pugui convertir. En funció dels permisos que tingui sobre el fitxer, el sistema convertirà aquest usuari a un dels tipus de l'aplicació. Independentment d'aquest tipus, fins que la presentació no estigui iniciada, el participant pot navegar lliurement per les diapositives.

Actors: usuari d'eyeOS

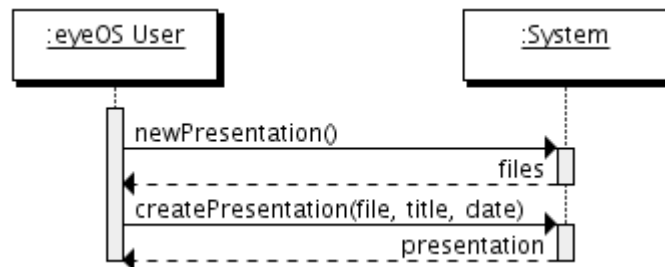
Activació: un usuari d'eyeOS vol obrir una presentació amb l'aplicació.

Escenari principal:

1. L'usuari indica al sistema que vol crear una presentació
2. El sistema mostra una llista de fitxers compatibles
3. L'usuari escull un fitxer, títol i data per la presentació.
4. El sistema dona d'alta una presentació, a partir del fitxer, amb el títol i data

indicats.

5. El sistema retorna una presentació.



Il·lustració 7: Diagrama de seqüència del sistema per crear una presentació

Obtenir

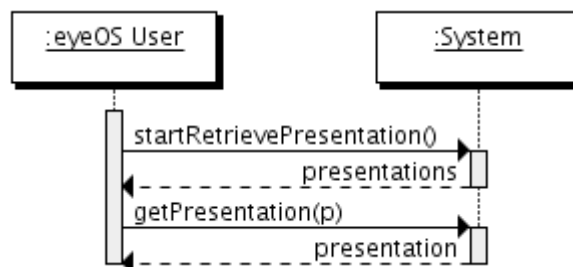
Un usuari de l'eyeOS ha de poder obtenir una presentació.

Actor: usuari de l'eyeOS

Activació: l'usuari de l'eyeOS vol obtenir una presentació

Escenari principal:

1. L'usuari indica al sistema que vol obtenir una presentació
2. El sistema mostra les presentacions
3. L'usuari n'escull una
4. El sistema retorna la presentació



Il·lustració 8: Diagrama de seqüència del sistema per obtenir una presentació

Actualitzar

Un Orador pot canviar el títol, data i hora d'una presentació. Els canvis es notifiquen als participants d'aquesta presentació.

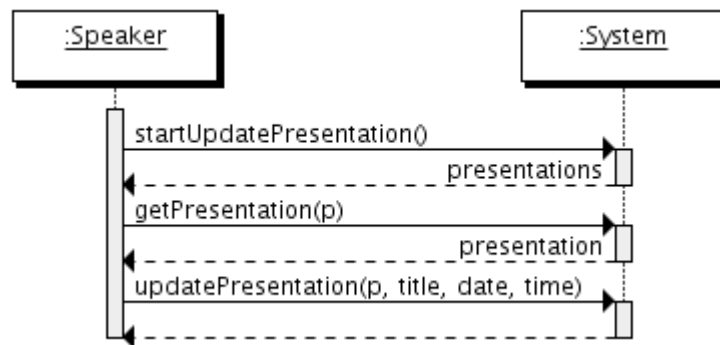
Actors: Orador

Activació: un Orador vol editar els detalls de la presentació.

Escenari principal:

1. L'orador escull una presentació
2. El sistema mostra el títol, data i hora enregistrats.
3. L'usuari indica al sistema el nou títol, data i hora.

4. El sistema enregistra els canvis.



Il·lustració 9: Diagrama de seqüència del sistema per actualitzar una presentació

Esborrar

Un Orador ha de poder esborrar una presentació.

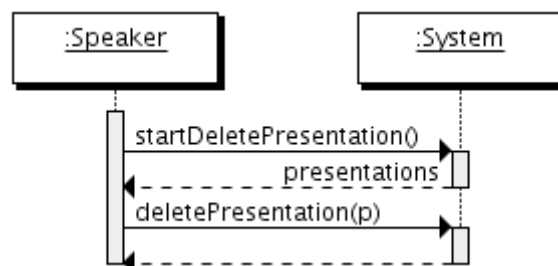
Actor: Orador

Activació: l'Orador vol donar de baixa una presentació

Qüestions obertes: aquest cas d'ús no serà implementat perquè col·lisiona amb problemes no solucionats de l'eyeOS. En particular, programes poden desar informació relacionada amb un fitxer que ja no existeix i introduir errors d'integritat. Per això, en aquesta versió de l'aplicació, una presentació es considera esborrada si no té participants excepte, com a molt, l'orador. No obstant, per formalitat i futura referència s'ha decidit deixar especificat aquest cas d'ús.

Escenari principal:

1. L'usuari indica al sistema que vol esborrar una presentació
2. El sistema mostra les presentacions
3. L'usuari indica al sistema quina presentació s'ha de donar de baixa
4. El sistema dóna de baixa la presentació i la seva informació associada.



Il·lustració 10: Diagrama de seqüència del sistema per esborrar una presentació

Mostrar diapositiva

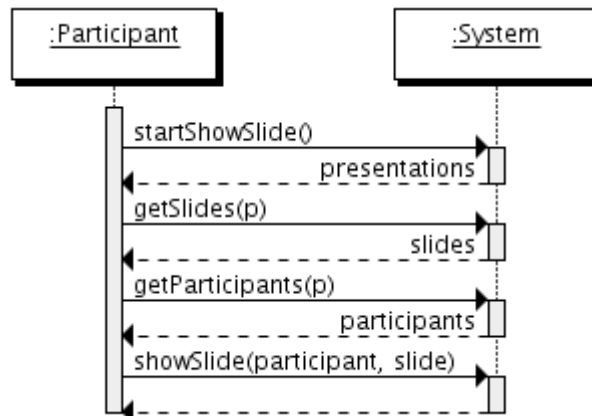
Mostra la diapositiva sol·licitada.

Actor: Participant

Activació: un participant sol·licita al sistema mostrar una diapositiva a un participant

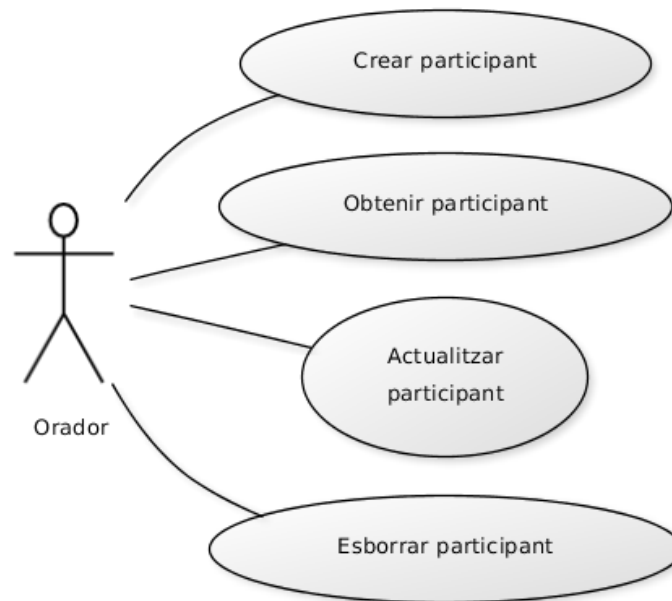
Escenari principal:

1. El participant indica al sistema que vol mostrar una diapositiva a un participant de la presentació.
2. El sistema mostra les presentacions
3. El sistema mostra les diapositives de la presentació escollida.
4. El sistema mostra els participants
5. El participant indica al sistema la diapositiva i el participant a qui mostrar-la.
6. El sistema mostra la diapositiva sol·licitada al participant escollit.



Il·lustració 11: Diagrama de seqüència del sistema per mostrar una diapositiva

4.3.2 Gestionar els participants



Il·lustració 12: Casos d'ús de gestionar participants

Crear

Un Orador afegeix un participant a la presentació.

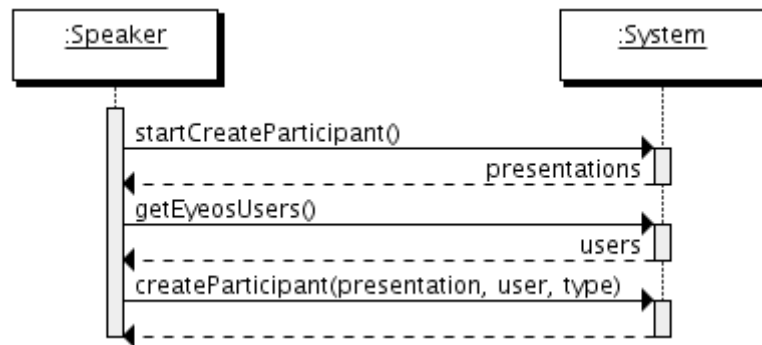
Actor: Orador

Activació: Un Orador vol afegir un participant a la presentació.

Qüestions obertes: en el disseny es pot decidir limitar els usuaris disponibles només als contactes o a un grup de treball determinat.

Escenari principal:

1. L'orador indica al sistema que vol afegir un participant.
2. El sistema mostra les presentacions.
3. El sistema mostra els usuaris de l'eyeOS
4. L'Orador indica al sistema les dades del nou participant: presentació en què participa, referència al contacte i tipus.
5. El sistema dona d'alta un participant en la presentació, amb contacte i tipus indicats.



Il·lustració 13: Diagrama de seqüència del sistema per crear un participant

Obtenir

Un Orador pot obtenir un participant de la presentació.

Actor: Orador

Activació: l'Orador vol obtenir un participant de la presentació.

Escenari principal:

1. L'Orador indica al sistema que vol veure els participants d'una presentació.
2. El sistema mostra les presentacions.
3. L'Orador escull una presentació
4. El sistema mostra els participants de la presentació.
5. L'orador indica al sistema l'identificador del participant
6. El sistema retorna el participant.

Nota: el diagrama és similar als anteriors.

Actualitzar

Un orador ha de poder canviar de quin tipus és un participant d'una presentació.

Actors: Orador

Activació: L'Orador vol actualitzar un participant d'una presentació.

Escenari principal:

1. L'orador indica al sistema que vol actualitzar les dades d'un participant d'una presentació.
2. El Sistema mostra les presentacions
3. L'Orador escull una presentació
4. El Sistema mostra els participants d'aquesta presentació
5. L'Orador indica al sistema el nou tipus del participant escollit.

Nota: el diagrama és similar als anteriors.

Esborrar

Un orador ha de poder esborrar un participant d'una presentació.

Actors: Orador

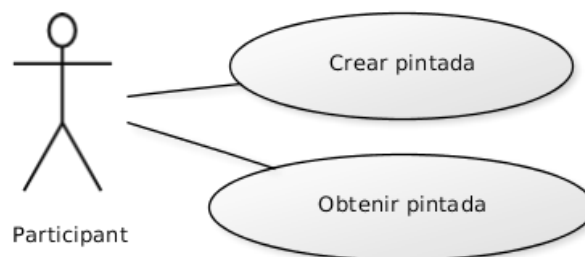
Activació: L'Orador vol esborrar un participant d'una presentació.

Escenari principal:

1. L'orador indica al sistema que vol esborrar un participant d'una presentació.
2. El sistema mostra les presentacions
3. L'orador escull una presentació
4. El sistema mostra els participants d'aquesta presentació
5. L'Orador indica al sistema el participant per donar de baixa.
6. El sistema dóna de baixa el participant escollit.

Nota: el diagrama és similar als anteriors.

4.3.3 Gestionar pintades



Il·lustració 14: Casos d'ús de gestionar pintades

Crear

Un participant ha de poder fer pintades sobre una diapositiva a mà alçada. El sistema decideix en funció dels permisos del participant si la pintada es pot compartir.

Actors: Participant

Activació: El Participant vol pintar sobre una diapositiva.

Escenari principal:

1. El Participant indica al sistema que vol pintar sobre una diapositiva.
2. El sistema mostra les presentacions del participant.
3. El Participant escull una presentació
4. El sistema mostra les diapositives d'aquesta presentació
5. El Participant indica al sistema la pintada, la diapositiva i la presentació.
6. El sistema enregistra la pintada.

Nota: el diagrama és similar als anteriors.

Obtenir

Un participant ha de poder obtenir una pintada d'una diapositiva.

Actors: Participant

Activació: El Participant vol obtenir les pintades d'una diapositiva.

Escenari principal:

1. El Participant indica al sistema que vol obtenir les pintades d'una diapositiva.
2. El sistema mostra les presentacions del participant.
3. El Participant escull una presentació
4. El sistema mostra les diapositives d'aquesta presentació
5. El Participant indica al sistema la diapositiva.
6. El sistema mostra les pintades de la diapositiva escollida.
7. El Participant indica al sistema l'identificador de la pintada
8. El sistema retorna la pintada

Nota: el diagrama és similar als anteriors.

Actualitzar

Un participant modifica una pintada que ha fet. Similar als anteriors.

Qüestions obertes: aquest cas d'ús és fora de l'abast del projecte.

Esborrar

Un participant esborra una pintada que ha fet. Similar als anteriors.

Qüestions obertes: aquest cas d'ús és fora de l'abast del projecte.

4.4 Contractes

Els contractes de les operacions del sistema poden ajudar a definir el comportament del sistema, descriuen el resultat de l'execució de les operacions del sistema en funció dels canvis d'estat dels objectes del domini.
[Larman]

Operació getPresentations(): presentations

Obté les presentacions relacionades amb el Participant que dispara el cas d'ús.

Precondicions: -

Postcondicions

1. presentations és un Set(Presentation) associat al Participant que dispara

el cas d'ús.

Sortida: presentations

Operació getPresentation(p: String): presentation

Obté la Presentation p

Precondicions

1. Existeix una Presentation amb file = p.

Postcondicions

1. presentation és una Presentation amb file = p.

Sortida: presentation

Operació start(p: Presentation)

Inicia la presentació p.

Precondicions

1. Existeix una Presentation p pel Participant que dispara el cas d'ús.
2. El Participant que dispara el cas d'ús és Speaker per la Presentation p.

Postcondicions

1. p.started = true

Sortida: -

Operació stop(p: Presentation)

Atura la presentació p.

Precondicions

1. Existeix una Presentation p pel Participant que dispara el cas d'ús.
2. El Participant que dispara el cas d'ús és Speaker per la Presentation p.

Postcondicions

1. p.started = false

Sortida: -

Operació newPresentation() : files

Inicia el cas d'ús crear una nova presentació.

Precondicions: -

Postcondicions

1. files és un Set(String) amb els noms dels fitxers disponibles per l'usuari que dispara el cas d'ús.

Sortida: files

Operació createPresentation(file: String, title: String, date: Date, time: Time)

Dóna d'alta una nova Presentation.

Precondicions

1. Existeix un fitxer amb nom *file*, l'usuari té suficients permisos per utilitzar-lo i és d'un format de presentacions compatible amb l'OpenOffice.
2. date+time formen una data > data actual.
3. No existeix una Presentation p amb p.speaker = usuari que dispara el cas d'ús i p.date = date+time.

Postcondicions

1. S'ha creat una instància de Presentation p amb p.date = date+time, title = title, started = false, i p.slide a partir del fitxer file.
2. presentation és p

Sortida: -

Operació startRetrievePresentation(): presentations

Inicia el cas d'ús obtenir una presentació.

Precondicions: -

Postcondicions

1. presentations és un Set(Presentation) associat al Participant que dispara el cas d'ús.

Sortida: presentations

Operació startUpdatePresentation(): presentations

Inicia el cas d'ús actualitzar una presentació.

Precondicions: -

Postcondicions

1. presentations és un Set(Presentation) associat al Participant que dispara el cas d'ús.

Sortida: presentations

Operació updatePresentation(p: String, title: String, date: Date, time: Time)

Actualitza les dades de la Presentation p.

Precondicions

1. Existeix una Presentation amb file = p.
2. date+time formen una data > data actual.
3. No existeix una Presentation pres amb pres.file = file, pres.speaker = usuari

que dispara el cas d'ús i `pres.date = date+time`.

Postcondicions

1. La Presentation `pres` té `pres.title = title`, `pres.date = date+time`

Sortida: -

Operació `startDeletePresentation()`: presentations

Inicia el cas d'ús obtenir una presentació.

Precondicions: -

Postcondicions

1. `presentations` és un `Set(Presentation)` associat al Participant que dispara el cas d'ús.

Sortida: `presentations`

Operació `deletePresentation(p: string)`

Inicia el cas d'ús obtenir una presentació.

Precondicions

1. Existeix una Presentation `pres` amb `pres.file = p`.

Postcondicions

1. S'ha donat de baixa la instància `pres` de Presentation, `pres.file = p`.

Sortida: -

Operació `startShowSlide()`: presentations

Inicia el cas d'ús mostrar diapositiva

Precondicions: -

Postcondicions

1. `presentations` és un `Set(Presentation)` associat al Participant que dispara el cas d'ús.

Sortida: `presentations`

Operació `getSlides(p: Presentation) : slides`

Obté les diapositives de la presentació `p`.

Precondicions

1. Existeix una Presentation `p`

Postcondicions

1. `slides` és un `Set(Slide) = p.slide`

Sortida: `slides`

Operació showSlide(p: Participant, s: Slide)

Inicia el cas d'ús mostrar diapositiva.

Precondicions

1. Existeix un Participant p en U.presentation.participant, on U és el participant que dispara el cas d'ús.
2. Existeix una Slide s en U.presentation.slide, on U és el participant que dispara el cas d'ús.

Postcondicions:

1. S'ha mostrat la diapositiva s al participant p

Sortida: -

Operació startCreateParticipant(): presentations

Inicia el cas d'ús crear participant.

Precondicions: -*Postcondicions*

1. presentations és un Set(Presentation) associat al Participant que dispara el cas d'ús.

Sortida: presentations

Operació getEyeOSUsers() : users

Obté els usuaris del sistema.

Precondicions: -*Postcondicions*

1. users és un Set(EyeOSUser) de tots els usuaris del sistema.

Sortida: users

Operació createParticipant(presentation: Presentation, user: EyeOSUser, type: String)

Dóna d'alta un participant en la presentació.

Precondicions

1. Existeix una Presentation presentation
2. Existeix un EyeOSUser user
3. type té valor «active» o «passive»
4. L'usuari que dispara el cas d'ús pertany al conjunt d'oradors de la presentació.

Postcondicions

1. S'ha donat d'alta una instància de Participant

2. S'ha associat amb presentation
3. S'ha associat amb user amb l'especialització type.

Sortida: -

Operació startRetrieveParticipants()

Inicia el cas d'ús obtenir els participants d'una presentació.

Precondicions: -

Postcondicions

1. presentations és un Set(Presentation) associat al Participant que dispara el cas d'ús.

Sortida: presentations

Operació getParticipants(p: Presentation) : participants

Obté els participants de la presentació p.

Precondicions

1. Existeix una Presentation p.

Postcondicions

1. participants = p.participant

Sortida: participants

Operació updateParticipant(p: Participant, type: String)

Actualitza les dades del participant p. En concret, en canvia la subclasse en funció de type.

Precondicions

1. Existeix un Participant p.
2. L'usuari que dispara el cas d'ús és un orador de la presentació a què pertany p.

Postcondicions

1. p és de tipus type.

Sortida: -

Operació deleteParticipant(p: Participant)

Dóna de baixa el participant p.

Precondicions

1. Existeix un Participant p.
2. L'usuari que dispara el cas d'ús és un orador de la presentació a què pertany p.

Postcondicions

1. S'ha donat de baixa la instància p del participant.
2. S'ha trencat l'associació amb la presentació p.pres.
3. S'ha trencat l'associació amb l'eyeOS User de p.eyeOSUser

Sortida: -

Operació startCreateDoodle() : presentations

Inicia el cas d'ús fer una pintada en una diapositiva.

Precondicions: -

Postcondicions

1. presentations és un Set(Presentation) associat al Participant que dispara el cas d'ús.

Sortida: presentations

Operació createDoodle(s: Slide, doodle: String)

Crea una pintada en una diapositiva

Precondicions

1. L'usuari que dispara el cas d'ús té permisos suficients per pintar sobre la diapositiva s.
2. Existeix una Slide s

Postcondicions

1. S'ha donat d'alta una instància de Doodle d amb d.path = doodle i d.slide = s
2. S'ha afegit d a s.doodle.

Sortida: -

Operació startRetrieveDoodle()

Inicia el cas d'ús obtenir una pintada

Precondicions: -

Postcondicions

1. presentations és un Set(Presentation) associat al Participant que dispara el cas d'ús.

Sortida: presentations

Operació getDoodles(s: Slide) : doodles

Obté les pintades de la diapositiva s

Precondicions

1. Existeix una Slide s per la presentació actual

Postcondicions

1. doodles és s.doodle

Sortida: doodles

Operació getDoodle(id: String) : doodle

Obté la pintada amb identificador id.

Precondicions

1. Existeix un Doodle d amb d.id = id

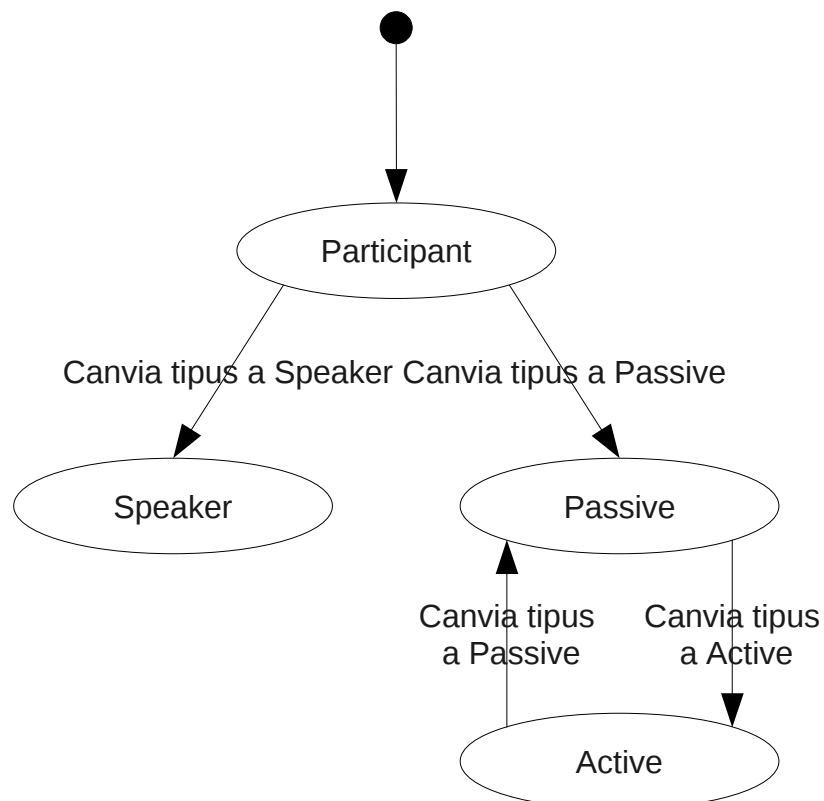
Postcondicions

1. doodle és un Doodle amb doodle.id = id.

Sortida: doodle

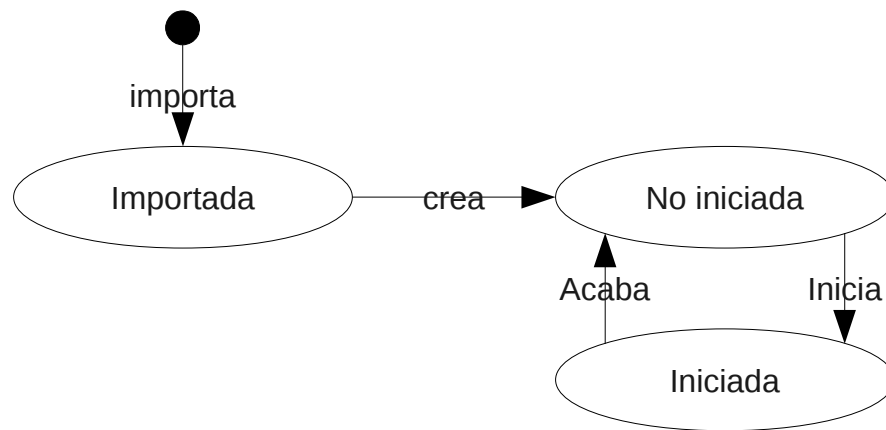
4.5 Diagrames d'estat

La Il·lustració 15 mostra la seqüència d'estats per què passa l'objecte Participant durant la seva vida en resposta a les accions que rep. Complementa la descripció de casos d'ús, on s'especifica quan es poden fer aquests canvis de subclasse.



Il·lustració 15: Diagrama d'estats dels tipus d'usuari

La Il·lustració 16 mostra els estats en què pot estar una presentació.



Il·lustració 16: Diagrama d'estats d'una presentació

5 DISSENY

El disseny de l'aplicació requereix un coneixement de la tecnologia que s'utilitzarà. Aquest capítol està dividit en dues parts

- Disseny de l'eyeOS
- Disseny de l'aplicació

S'ofereix una descripció del disseny de l'eyeOS que complementa el capítol introductori en què s'explica la plataforma des del punt de vista de l'usuari final.

De l'aplicació d'aquest projecte se'n dona una descripció del disseny en profunditat.

En aquest estadi s'han pres dues decisions:

1. Limitar a una les presentacions en curs per cada instància del programa. S'ha fet per qüestions d'usabilitat òbvies: una persona no hauria de veure més d'una presentació a la vegada perquè el canal de comunicació és extern a l'aplicació i interferirien sense un control precís de l'usuari.
2. Permetre només un orador per presentació, que és el mateix que el propietari del fitxer. S'ha fet així per simplificar el desenvolupament i aprofitar les funcions de l'eyeOS en gestió dels fitxers. No obstant, res no impedeix ampliar això en el futur.

5.1 L'entorn: eyeOS⁷

5.1.1 Vista estàtica

5.1.1.1 Introducció

L'eyeOS té una base de codi considerablement gran i està fortament acoblat al servidor en què s'executa. Utilitza una biblioteca externa, Qooxdoo, perquè els desenvolupadors puguin utilitzar orientació a objectes en el client (amb JavaScript). L'eyeOS ofereix una API rica tant en el client com en el servidor. Aquest capítol és la visió estàtica general del sistema. No entra en detalls com el diagrama de classes. La vista estàtica mostra quines de classes consisteix el

⁷ Bibliografia [Gamonal]

sistema i les relacions entre elles.

5.1.1.2 L'eyeOS és una aplicació web

A software system based on technologies and standards of the World Wide Web Consortium (W3C) that provides Web specific resources such as content and services through a user interface, the Web browser [Kappel]

L'eyeOS és una aplicació web gran que interacciona amb uns quants components reusables. Està construïda amb PHP i JavaScript i les dades s'enregistren amb un SGBD, preferiblement MySQL. No és un servei web ni una pàgina web.

Les característiques d'una aplicació web⁸ són presents tant en el disseny de l'eyeOS com en el de les aplicacions que hi funcionen. Són:

- *Ús intensiu de la xarxa.* Cal minimitzar crides remotes i estar familiaritzat amb patrons com la façana remota, proxy remot, DTO⁹ i RPC¹⁰.
 - L'eyeOS utilitza sovint un patró *Manager*¹¹ per alguns components (p. ex. els contactes) per aconseguir-ho.
- *Abast global i factors inesperats.*
 - L'eyeOS sovint s'adapta als clients però a vegades no es pot saber amb antelació i precisió quins i quants seran els usuaris (p. ex. <http://try.eyeos.org>) Poden aparèixer alguns problemes: concurrència, pics, càrrega del sistema... Això no és responsabilitat del desenvolupador d'una aplicació, però sí dels desenvolupadors del nucli de l'eyeOS i de l'administrador del sistema.
- *Competència i cerca dels factors diferenciadors.*
- *Espai i temps* (internacionalització i disponibilitat)
- *És hipertext:* no és lineal, hi ha diferents maneres de navegar i ha d'estar centrada en els continguts.
- *Aspecte i comportament.* Usabilitat, accessibilitat i estètica més propers a una aplicació d'escriptori que a un lloc web.
- *Seguretat.* És important no només per les dades enregistrades, sinó també en les comunicacions.

8 [DSBW]

9 Data Transfer Object

10 Remote Procedure Call

11 Vegeu capítol PresentationManager (pàg 81)

5.1.1.3 L'eyeOS i les seves aplicacions són distribuïts

Un programa típicament distribuït és una aplicació web o un sistema RPC. L'eyeOS és precisament una aplicació distribuïda i satisfà la majoria de les característiques que tenen aquests sistemes [Emmerich]:

- Diversos components autònoms.
- No tots els usuaris comparteixen els mateixos components.
- Els recursos podrien no ser accessibles.
- El programa pot executar-se en processos concurrents en diferents processadors.
- Diversos punts de control i de fallada.

Aquesta arquitectura té avantatges:

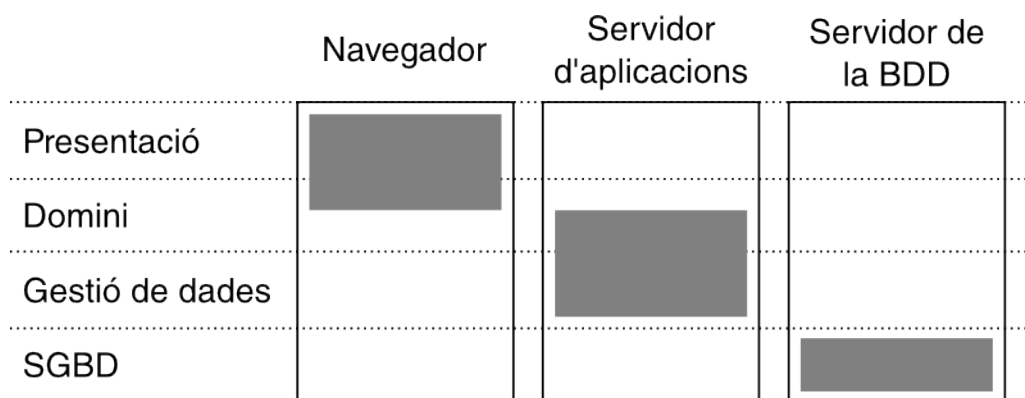
- Permet compartir recursos.
- Promou l'obertura.
- Concurrència.
- Escalabilitat.
- Tolerància a fallades.
- Transparència

L'eyeOS es pot utilitzar en tot tipus d'organitzacions que necessitin compartir dades de manera concurrent sense pensar en si serà massa pesat. La infraestructura que utilitzi l'eyeOS pot créixer i decreixer d'acord amb les necessitats del client i de manera transparent pel desenvolupador d'aplicacions. Aquest no s'ha de preocupar pel maquinari, ni per la concurrència ni per migracions ni escalabilitat. És transparent.

Quan s'escriu una aplicació per l'eyeOS no cal preocupar-se d'algunes coses que ja tracta el sistema, però cal tenir present que és distribuïda: cal minimitzar crides al servidor i mai no s'ha de confiar en les dades que introdueix l'usuari. La xarxa pot haver caigut, els paquets poden haver estar intervinguts, les dades poden arribar incompletes... Cal tenir sempre molt clar on és la frontera del sistema, on acaba el node local.

5.1.1.4 Arquitectura de l'eyeOS

L'eyeOS té una arquitectura en 3 capes i està típicament distribuïda en 3 nodes, com indica la Il·lustració 17.



Il·lustració 17: Distribució de capes en els nodes

El navegador (node 1) té la capa de presentació i part del domini. El servidor d'aplicacions (p. ex. l'Apache, en el node 2), part del domini i de la capa de gestió de dades. El servidor de la base de dades (node 3) allotja el SGBD. És bastant comú (i així s'ha fet en aquest projecte) que es fusionin el servidor d'aplicacions i la base de dades en una sola màquina en una simple instal·lació LAMP.

Els desenvolupadors d'aplicacions per l'eyeOS haurien d'utilitzar el patró d'orientació a objectes. Com es pot veure en la Il·lustració 17, part del programa s'executa en el navegador i part en el servidor d'aplicacions.

D'ara endavant en aquest document, el navegador s'anomenarà *client* i el servidor d'aplicacions, simplement *servidor*. Cal dissenyar l'aplicació (vista de desenvolupament) com si fos un programa d'escriptori tradicional i després decidir quines classes pertanyen a quin node (vista de desplegament). Es tracta de determinar la frontera del sistema.

En l'eyeOS les grans diferències (però no les úniques ni necessàriament l'únic criteri a seguir) que ajuden a determinar la frontera del sistema són:

- Els objectes del costat del client estan vius fins que l'usuari refresca la pàgina.
- Els objectes del costat del servidor estan vius només durant la transacció, mentre s'utilitzen.

Arquitectura en tres capes

L'eyeOS té una arquitectura en tres capes. S'ofereixen eines al desenvolupador en cadascuna.

- *Presentació*

S'allotja al navegador. Qooxdoo permet al desenvolupador escriure la interfície gràfica com si fos Swing, AWT o GTK. És senzill, amigable i hi ha una comunitat al darrere.

- *Domini*

- Client: Qooxdoo permet al desenvolupador fer servir orientació a objectes. JavaScript és un llenguatge *funcional* i orientat a objectes, però Qooxdoo ho «apedaça».
- Servidor: el codi s'escriu en PHP. S'utilitza l'API de PHP i els serveis, frameworks i l'API de l'eyeOS. Es busca sempre escriure codi reusable i evitar reinventar la roda. Patrons habituals: façana remota, proxy remot, controlador d'aplicació.

- *Gestió de dades*

DAO (*Database Abstraction Layer*) i PEAR. En el moment d'escriure aquest document, el DAO és massa dèbil per moltes tasques i cal utilitzar PEAR directament. Un patró comú aquí és DTO.

- *Emmagatzematges de dades*

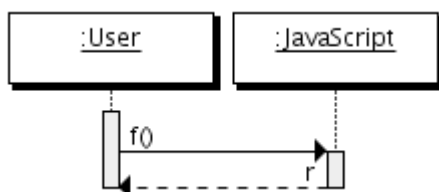
Típicament el MySQL. Es pot suportar DB2, XML, PostgreSQL... i qualsevol SGBD pel qual es pugui escriure un *provider*. En el moment d'escriure aquest document, el DAO és massa dèbil per poder utilitzar bé aquests *providers*.

5.1.2 Vista dinàmica

5.1.2.1 Visió general

Els objectes mostrats en la vista estàtica es comuniquen entre ells cridant les operacions que ofereixen. En aquest capítol s'expliquen les dues interaccions més freqüents: una crida local i una de remota.

5.1.2.2 Crida local



Il·lustració 18: Crida local

L'usuari dispara un esdeveniment que crida el mètode *f* d'un fitxer JavaScript. Normalment l'usuari interacciona amb la classe de la GUI que delega l'operació en els altres objectes.

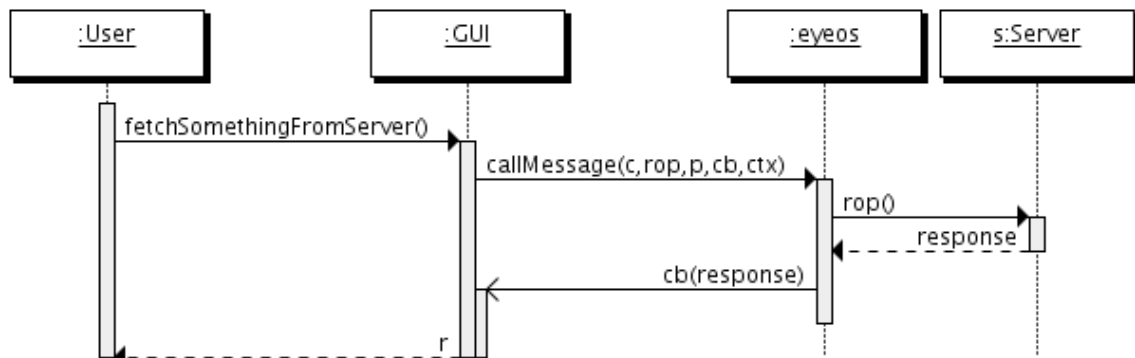
La classe principal una classe de Qooxdoo

anomenada *eyeos.application.laVostraApp* .

En el codi és *myObj.f()*;

5.1.2.3 Crida remota

Una de les característiques d'una aplicació distribuïda és que utilitza la xarxa de manera intensa. Això es controla en les crides remotes.



Il·lustració 19: Crida remota

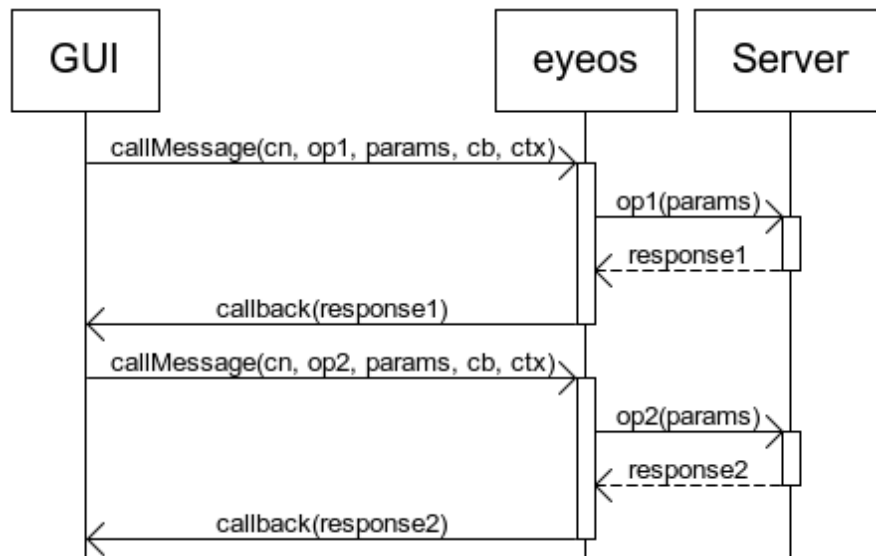
En la Il·lustració 19: Crida remota s'ha simplificat la seqüència. L'única classe del client és la GUI, la principal.

1. L'usuari dispara un cas d'ús per obtenir dades del servidor.
2. La GUI crida `eyeos.callMessage` amb els paràmetres:
 - *c*: el checknum de l'aplicació (`this.checknum`)
 - *rop* el nom de l'operació del servidor (`remote operation`).
 - *p* un mapa dels paràmetres que té l'operació remota.
 - *cb*: JavaScript és asíncron. El client demana quelcom al servidor i quan s'hagi acabat de processar la petició el servidor crida la funció *cb*, que en aquest cas és al client, i emplena un paràmetre (`response`) amb la resposta.
 - *ctx* és habitualment *this*. JavaScript necessita saber qui és el pare.
3. L'objecte `eyeos` delega `callMessage` i executa la petició al servidor.
4. El servidor executa la petició. Quan ha acabat, prepara una resposta (habitualment una matriu associativa que l'eyeOS converteix a Json) i la passa com a paràmetre a la funció de callback.

La resposta del servidor no cal que sigui de granularitat fina. Pot ser gruixuda de manera que no només s'obtingui la informació estrictament necessària. Això

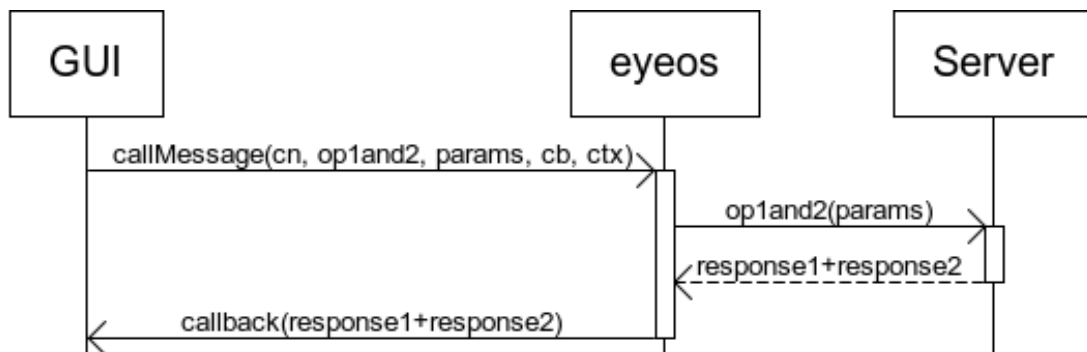
s'hauria de fer per minimitzar crides remotes de manera raonable.

Cal agrupar crides remotes on sigui possible. Compareu les il·lustracions 20 i 21.



Il·lustració 20: Dues crides consecutives amb respostes diferents

S'agrupen i s'obté un DTO o una tupla.



Il·lustració 21: Dues crides agrupades

Aquesta tupla pot ser, per exemple, una matriu associativa. La classe eyeos la pot convertir a Json perquè estigui a punt per fer servir en la crida a *callback*.

5.2 L'aplicació del projecte

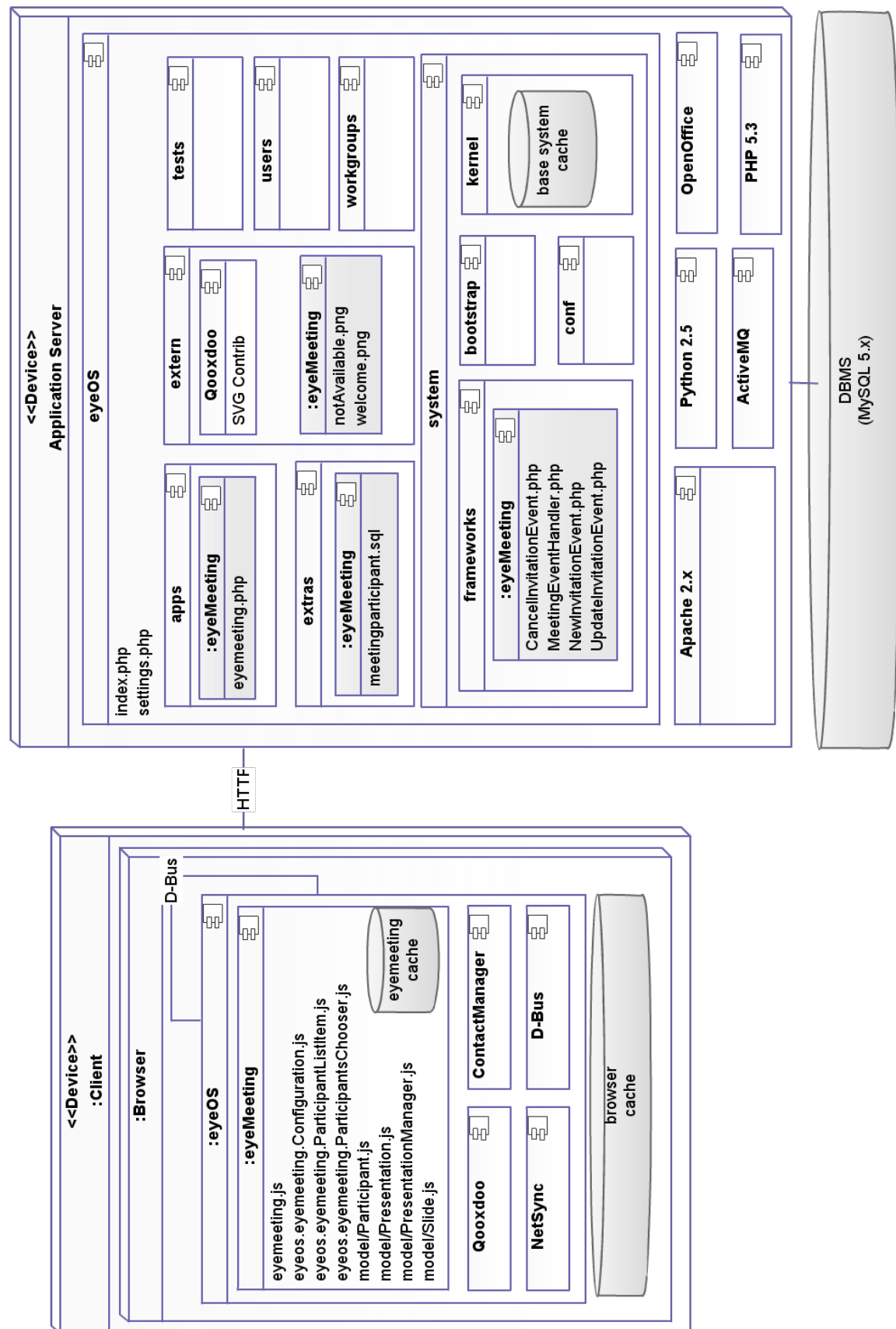
L'aplicació ha d'encaixar en l'entorn descrit. En les següents seccions s'ofereix una descripció en profunditat de com està dissenyada i per què s'ha fet així.

5.2.1 Arquitectura física

La vista de desplegament fa referència al maquinari en què s'instal·larà l'aplicació. Aquesta funciona dins l'eyeOS, que com es diu en el capítol 5.1.1.3. està distribuït, a grans trets, entre client i servidor. L'aplicació d'aquest projecte també,

tot i que alguns programes més senzills en l'eyeOS funcionen íntegrament en un node.

En la Il·lustració 22 apareix cada component de l'aplicació en el seu node. Per facilitar-ne la lectura els components exclusius de l'aplicació estan marcats en gris.



Il·lustració 22: Vista de desplegament de l'eyeMeeting

Hi apareixen també els components del nucli de l'eyeOS perquè sense ells l'aplicació no funcionaria i s'utilitzen contínuament.

L'aplicació té una part en el client, escrita en JavaScript, i una part en el servidor, en PHP.

S'ha procurat tenir el màxim d'objectes possibles en el client perquè es mantinguin vius durant tota l'execució del programa i per qüestions de rendiment.

Client

- *eyemeeting.js*
Interfície gràfica de la finestra principal.
- *eyeos.eyemeeting.Configuration.js*
Interfície gràfica de la finestra de configuració (títol, data i hora de la presentació).
- *eyeos.eyemeeting.ParticipantsChooser.js*
Interfície gràfica de la finestra per gestionar els participants.
- *model/Presentation.js*
Conté la classe que representa una presentació. Relaciona aquesta entitat amb els fitxers necessaris, n'enregistra el títol, etc.
- *model/Slide.js*
Similar a l'anterior.
- *model/Participant.js*
Similar a l'anterior.
- *model/PresentationManager.js*
Conté la classe que centralitza les comunicacions amb el servidor.

La memòria cau de l'eyeMeeting en el client s'ha aconseguit per 3 vies:

- Memòria cau de Qooxdoo per l'element `qx.ui.Image` (que utilitzen les miniatures)
- Enregistrament serialitzat de les imatges de les diapositives en la seva representació dins l'element SVG.
- `PresentationManager`.

Servidor

Una aplicació senzilla en l'eyeOS hauria de tenir només fitxers en la seva carpeta

dins /apps/. En el cas de l'eyeMeeting ha fet falta utilitzar i estendre components fora de la carpeta d'/apps/, si bé és cert que ja estan pensats perquè s'hi afegixin elements.

- */apps/eyemeeting/eyemeeting.php*

Façana de l'aplicació.

- *Frameworks*

Un framework en l'eyeOS ofereix uns mètodes a totes les aplicacions. Existia el framework Events, que gestiona els esdeveniments entre usuaris. Per exemple les invitacions per afegir un contacte, invitacions per compartir un fitxer, etc. Amb aquesta aplicació s'ha extès per poder enviar invitacions i notificacions als usuaris:

- Invitació a la presentació
- Cancel·lació
- Actualització dels detalls (títol, data, hora, fitxer)

- *extras/meetingparticipant.sql*

En aquesta carpeta s'hi desen fitxers que no es necessiten per l'execució de l'eyeOS però sí per la definició d'alguns components. meetingparticipant.sql conté la definició de l'única taula de la base de dades que fa servir eyeMeeting¹².

- *extern/images/meeting i extern/js/qx.js*

En extern s'hi desen els recursos accessibles des del navegador, com per exemple imatges de les aplicacions. També conté els fitxers que s'han de servir al client per fer funcionar l'eyeOS (però no el JavaScript de les aplicacions). Per exemple, els fitxers de client de Qooxdoo. En el cas d'eyeMeeting ha fet falta utilitzar el contrib SVG. Per poder-ho fer, s'ha hagut de «compilar¹³» el Qooxdoo i substituir el fitxer extern/js/qx.js pel nou.

La resta de components del servidor no formen part de l'abast d'aquest projecte però estan degudament documentats en el manual de desenvolupadors¹⁴ de l'eyeOS 2.x.

12 Vegeu el capítol 5.2.2.4 Disseny de la capa d'accés a dades

13 Terminologia pròpia de Qooxdoo. Es refereix a resoldre dependències, incloure components i comprimir el fitxer final. JavaScript és un llenguatge interpretat.

14Consulteu la Bibliografia i referències

5.2.2 Arquitectura lògica: disseny en capes

Desenvolupar una aplicació per l'eyeOS és, simplificadament, escriure'n un component. Cal ser coherent amb l'arquitectura en 3 capes de la plataforma i integrar-hi bé l'aplicació.

5.2.2.1 Tecnologies utilitzades en cada capa

En general una aplicació ha de tenir JavaScript i PHP. Es pot utilitzar el Flash, per exemple, però han d'estar encapsulats en algun component de Qooxdoo. En aquest cas s'ha utilitzat SVG per enregistrar les pintades, com s'explica en els capítols Disseny intern (pàg. 58) i 6.2.3 Implementació - SVG (pàg. 106)

- *Presentació*
JavaScript amb Qooxdoo 1.3 i el contrib SVG 0.1.
- *Domini*
JavaScript amb Qooxdoo i PHP 5.3.3 amb l'API de l'eyeOS.
- *Gestió de dades*
PHP amb l'API de l'eyeOS sempre que sigui possible.
- *Dades*
MySQL 5.x

JavaScript

El JavaScript és el llenguatge de la xarxa. Va començar com una manera de manipular uns pocs elements en una pàgina web (com imatges o camps de formularis) però ha crescut notòriament. A més dels scripts del costat del client, a dia d'avui es pot utilitzar per programar per una gran varietat de plataformes. Es pot escriure codi del costat del servidor (amb .Net o Node.js), aplicacions d'escriptori que funcionen en tots els sistemes operatius i extensions d'aplicacions (per exemple, pel Firefox o pel Photoshop), aplicacions mòbils i scripts de línia d'ordres.

El JavaScript és un llenguatge poc comú. No té classes i les funcions són objectes de primer ordre que s'utilitzen per moltes coses. [...] En els darrers anys llenguatges com el Java o el PHP han començat a afegir característiques com les clausures i les funcions anònimes, que han estat disponibles amb JavaScript des de temps enrere. [Stefanov]

És un llenguatge orientat a objectes i només té cinc tipus primitius (number, string, boolean, null, undefined). Un objecte és simplement una col·lecció de propietats donades, una llista de parells clau-valor. Algunes propietats poden ser funcions i

llavors s'anomenen mètodes.

No té classes, només objectes. Es crea un objecte en blanc (que no té propietats «pròpies» -- *own properties*) i s'hi afegeixen els membres convenients. Precisament una de les regles generals en el llibre de Gang of Four és «preferiu compondre objectes a l'herència».

JavaScript no té herència, sinó que utilitza prototips per simular-la i reusar codi. Un prototip és un objecte i cada funció que es crea té automàticament una propietat anomenada *prototype* que apunta a un objecte blanc.

SVG

L'SVG¹⁵ és un format popular lliure de drets d'autor desenvolupat i mantingut pel W3C SVG Working Group. És un llenguatge modelat per descriure gràfics vectorials bidimensionals estàtics o dinàmics en XML.

Com que són fitxers XML, les imatges de SVG es poden crear i editar amb qualsevol editor de text o amb programes de dibuix especialitzats, com l'Inkscape.

En el cas de l'aplicació d'aquest projecte, l'usuari pot pintar en el llenç com si fos un programa de dibuix i el sistema ho enregistra en format SVG.

Qooxdoo

Qooxdoo és un *framework* en JavaScript que permet crear aplicacions per diverses plataformes. Segueix un model de programació orientada a objectes que permet construir aplicacions riques i interactives. És codi obert subjecte a diverses llicències lliures i té una comunitat força activa a dia d'avui [QxDoc]

Amb Qooxdoo s'amaga la naturalesa sense classes de JavaScript, que incomoda alguns desenvolupadors, i es pot escriure un programa d'una manera molt similar a com seria una d'escriptori.

Per l'aplicació d'aquest projecte es va començar amb Qooxdoo 1.2 (eyeOS 2.3) i es va actualitzar a Qooxdoo 1.3 (eyeOS 2.5).

¹⁵ Consulteu la Bibliografia i referències

SVG contrib

A més a més de Qooxdoo ha fet falta el contrib SVG 0.1. Encapsula algunes de les funcions més habituals en treballar amb SVG i l'integra en Qooxdoo com si fos un *widget* més.

PHP

És un llenguatge de programació interpretat que s'utilitza per a generar pàgines web de forma dinàmica. S'executa al costat del servidor i envia només HTML al client. És orientat a objectes i està extremadament modularitzat, és multiplataforma, permet una interacció fàcil amb bases de dades, és lliure i està molt ben documentat.

Tota aplicació d'eyeOS necessita com a mínim un fitxer PHP al servidor que, com a mínim, serveixi els fitxers JavaScript al client.

MySQL 5.x

L'eyeOS utilitza com a sistema gestor de bases de dades MySQL 5.x. És possible utilitzar-ne d'altres però a dia d'avui és una tasca complexa i sense garanties de bon resultat pel disseny de la capa d'abstracció de dades. MySQL pot usar diferents motors a l'hora d'implementar la base de dades. En el cas de la base de dades de l'eyeOS s'utilitza l'InnoDB, que permet tractar la integritat referencial millor que amb el motor típic MyISAM.

Per contra, no suporta transaccions, *rollbacks* ni subconsultes, però és un SGBD molt popular en entorns Linux-Apache-PHP, que és precisament el que necessita l'eyeOS i el que espera la comunitat.

5.2.2.2 Disseny de la capa de presentació web

El disseny de la capa de presentació té dues parts: el disseny extern i l'intern. En el primer es defineix com ha de ser la interfície mentre que el segon defineix com interactua la capa de presentació amb la seva capa adjacent inferior.

Habitualment en una aplicació web es faria servir l'extensió web de l'UML per dissenyar aquesta part. No obstant, una aplicació en l'eyeOS és un cas particular: s'imita una aplicació d'escriptori i per tant l'extensió web de l'UML no és del tot útil.

La interfície s'ha fet seguint algunes directrius de l'equip de disseny i usabilitat. Tot i ser senzilla, s'ha volgut donar el control complet a l'usuari, evitar que hagi de memoritzar res i que tot sigui consistent tant gràficament com lingüísticament [Pressman].

Disseny extern

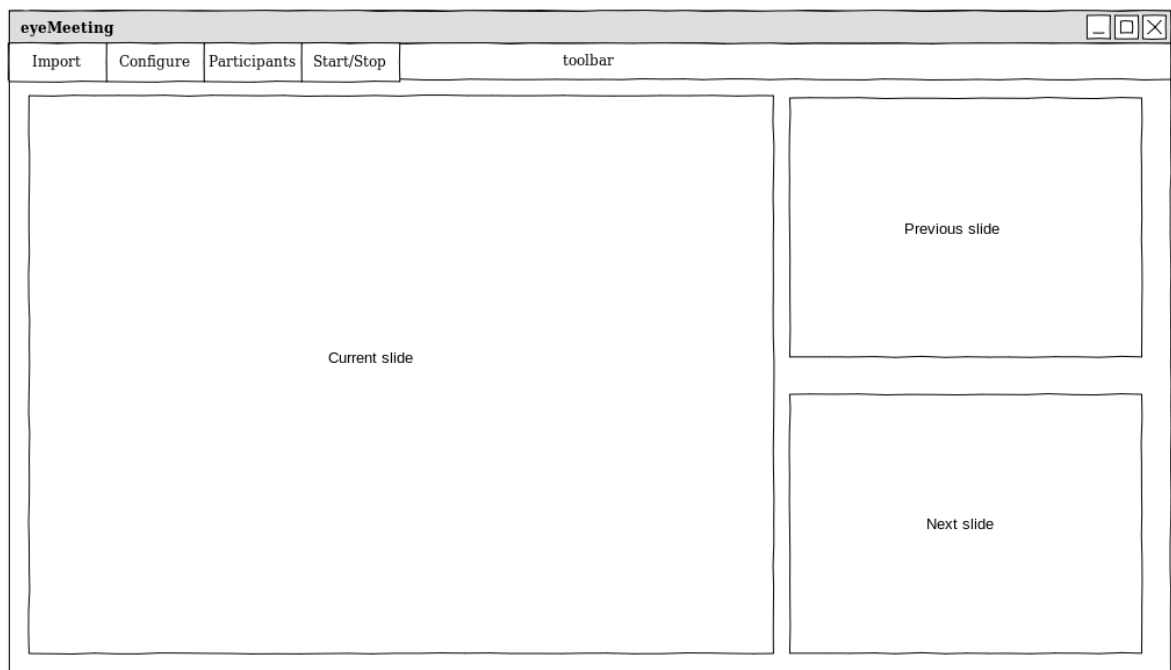
A partir de l'especificació es pot crear una interfície. En aquesta secció es defineixen els mecanismes d'interacció amb què l'usuari pot fer peticions al sistema i els mecanismes de presentació de la informació amb què es mostren a l'usuari els resultats de les seves peticions.

La vista de l'orador és més completa que la dels altres participants. Per qüestions de temps s'ha prescindit d'alguns detalls d'usabilitat i s'han deixat com tasques futures:

- No es pot escollir l'eina per pintar sobre les diapositives. Només n'hi ha una.
- No hi ha cap zona on s'indiqui a l'usuari el seu estat.
- No s'indica a l'usuari que el programa processa informació durant accions de llarga durada (p. ex. un rellotge o barra de progrés en importar la presentació).

En les figures següents es mostren totes les vistes del programa amb un nom pels seus controls i una descripció d'alt nivell de les funcions que fan.

La Il·lustració 23 és la pantalla principal per un orador.

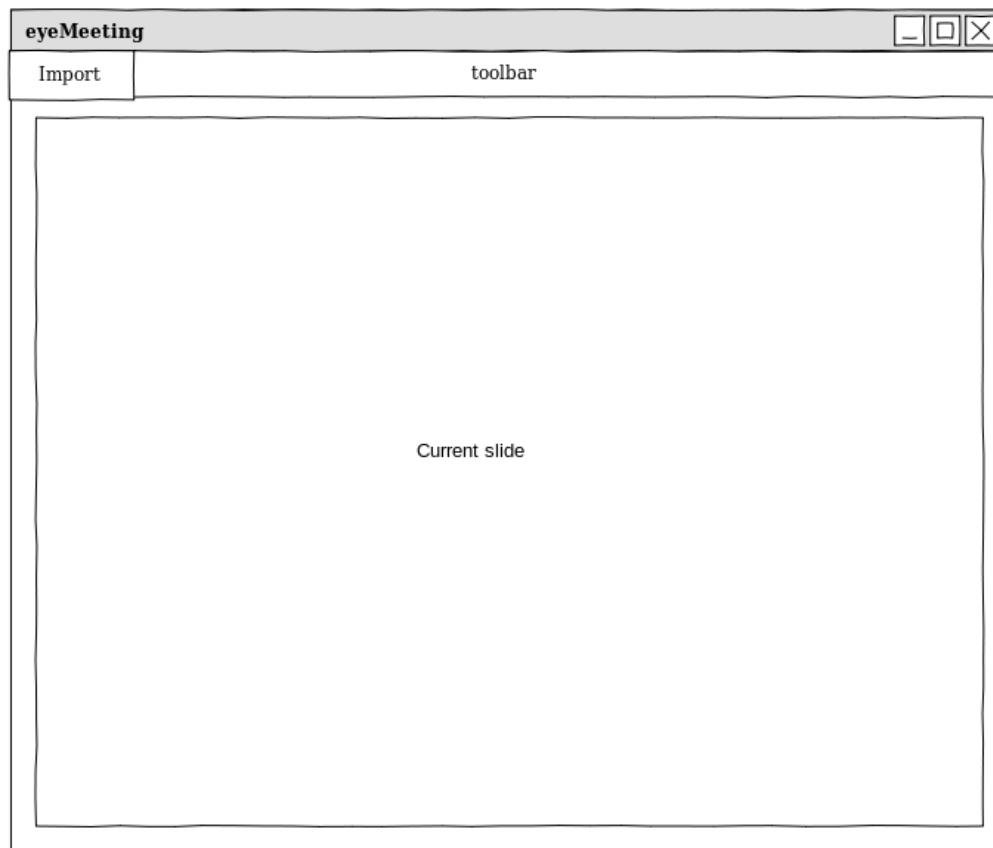


Il·lustració 23: Wireframe de la finestra principal (vista d'orador)

Els controls de què disposa són:

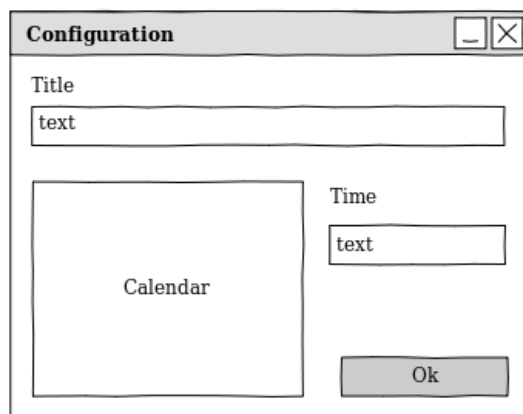
- *Import* (botó): obre un quadre de diàleg del sistema per crear una presentació en l'aplicació a partir d'un fitxer en format odp, ppt o pptx. Quan la presentació s'ha importat, s'inhabilita.
- *Configuration* (botó): obre una finestra modal perquè un orador pugui entrar el títol, data i hora previstes d'inici de la presentació.
- *Participants* (botó): obre una finestra modal perquè un orador pugui gestionar els participants de la presentació.
- *Start/Stop* (botó): commuta el comportament d'iniciar o acabar una presentació.
- *Next* i *Previous Slide*: són miniatures de les diapositives següent i anterior, on l'orador pot fer clic, per moure's entre elles. Són d'una mida suficientment gran per poder-ne llegir el contingut principal i ajudar l'orador a mantenir el fil de la presentació.

La pantalla principal per un oient (Il·lustració 24), que és el punt d'entrada comú per qualsevol usuari de l'eyeOS, disposa només d'un botó a la barra d'eines, amb el mateix comportament que en la vista d'orador.



Il·lustració 24: Wireframe de la pantalla principal (vista d'oient)

La vista de configuració permet a l'orador actualitzar detalls de la presentació



Il·lustració 25: Wireframe de la finestra de configuració

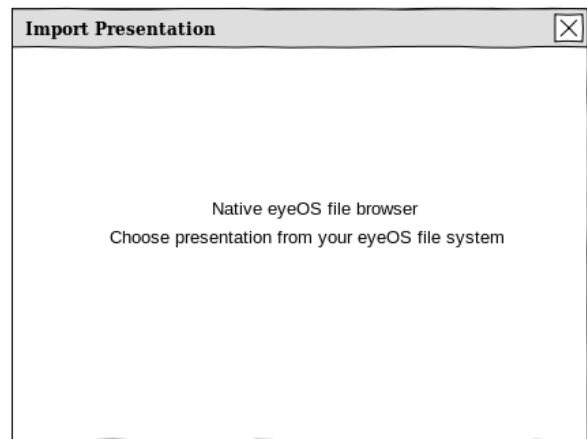
Com era d'esperar, té:

- *Title* (quadre de text): per entrar-hi el títol limitat a una mida raonable.
- *Time* (quadre de text): per entrar-hi l'hora. Qooxdoo no té encara un widget per entrar l'hora.
- *Calendar* (calendari): un *widget* de Qooxdoo amb un calendari per evitar formats incorrectes d'entrada en aquest camp.

- *Ok* (botó): per trametre els canvis. Es fan les validacions de dades corresponents i en cas d'error es marquen els camps afectats.

La finestra d'importació (Il·lustració 26) és un quadre de diàleg de l'eyeOS anomenat FileChooser. Permet navegar pels fitxers de l'usuari, els compartits i els grups de treball tot filtrant per format.

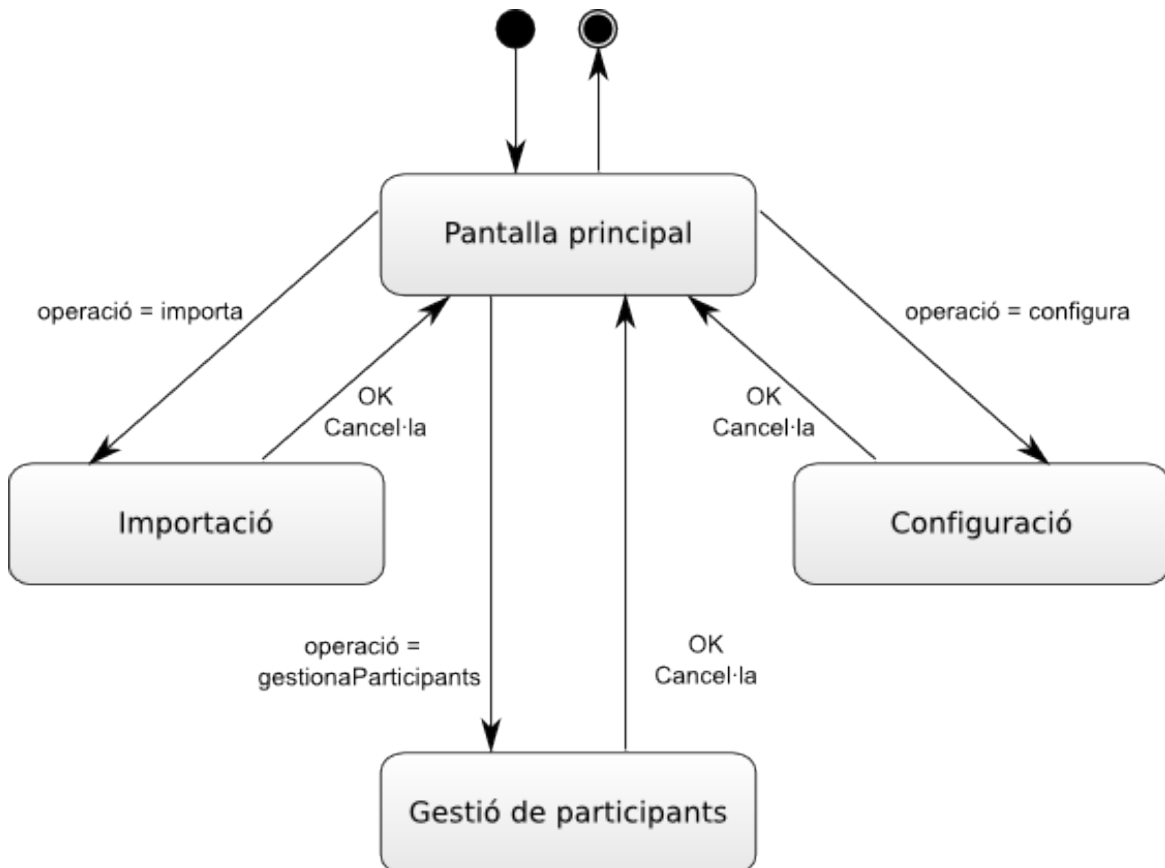
En prémer Ok es tramet el nom de fitxer i l'aplicació fa la importació.



Il·lustració 26: Wireframe del navegador de fitxers

Els mapes navegacionals següents defineixen el comportament del sistema i les vistes en cada cas d'ús.

La Il·lustració 27 és el mapa navegacional d'àmbit de sistema; una manera de mostrar els canvis de vistes de manera més general.



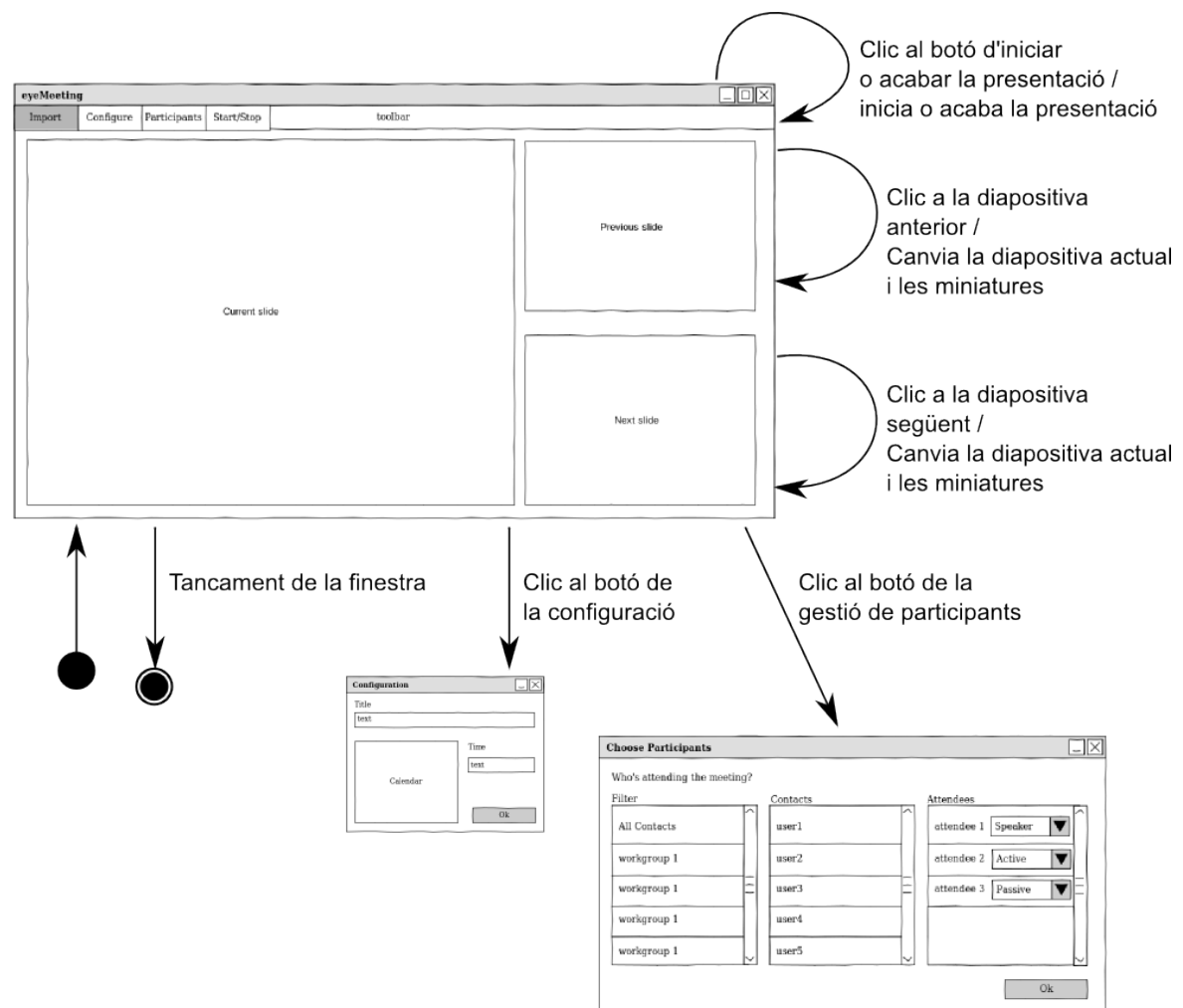
Il·lustració 27: Mapa navegacional d'àmbit de sistema

Es pot veure com canvien les pantalles en detall si es fan els mapes navegacionals per casos d'ús. Els de les següents pàgines han estat agrupats per brevetat i comoditat de lectura sempre que s'ha pogut. La sintaxi és l'habitual

en un diagrama d'estat: els nodes són estats, les arestes indiquen les transicions i el text de les arestes té la forma *esdeveniment [condició] / acció*. Quan no s'indica acció se sobreentén que es tracta simplement d'un canvi de pantalla.

Casos d'ús de gestionar la presentació

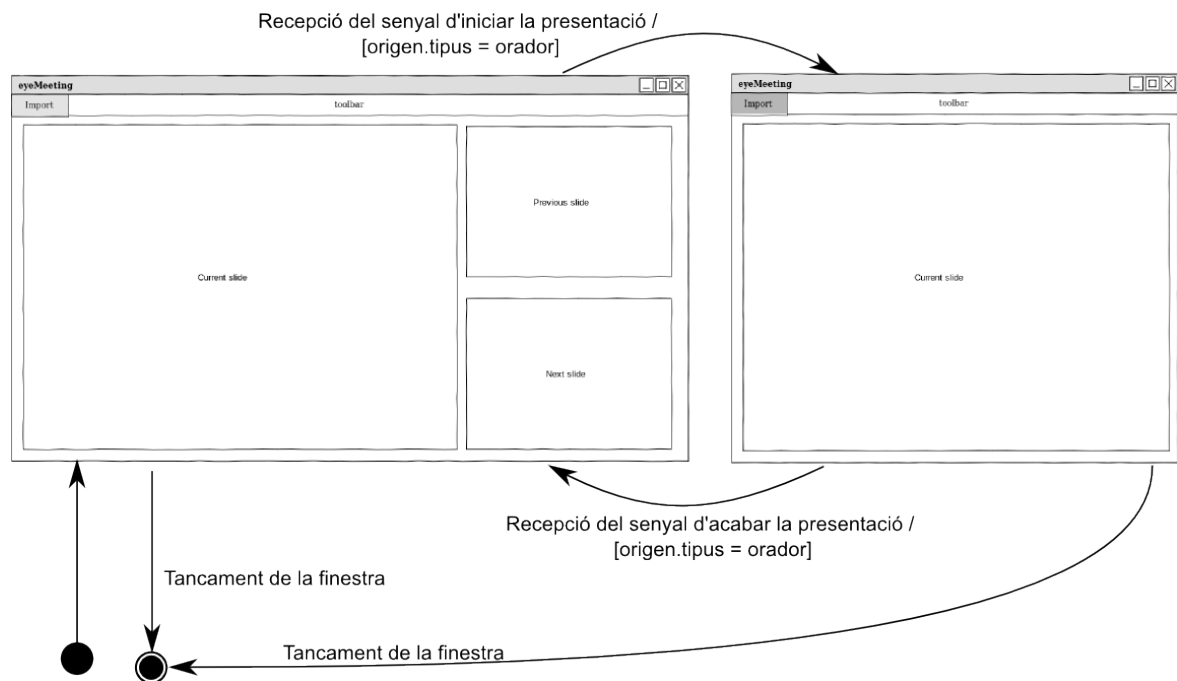
Els casos d'ús d'iniciar i acabar una presentació juntament amb mostrar una diapositiva, vista d'orador. En l'especificació el cas de mostrar diapositiva estava assignat a un Participant, que tant pot ser Orador com Oient. S'ha decidit limitar la interfície gràfica perquè un usuari sense els permisos adequats (que no sigui oient) no pugui demanar al sistema canviar la diapositiva si la presentació ja ha començat (Il·lustració 28).



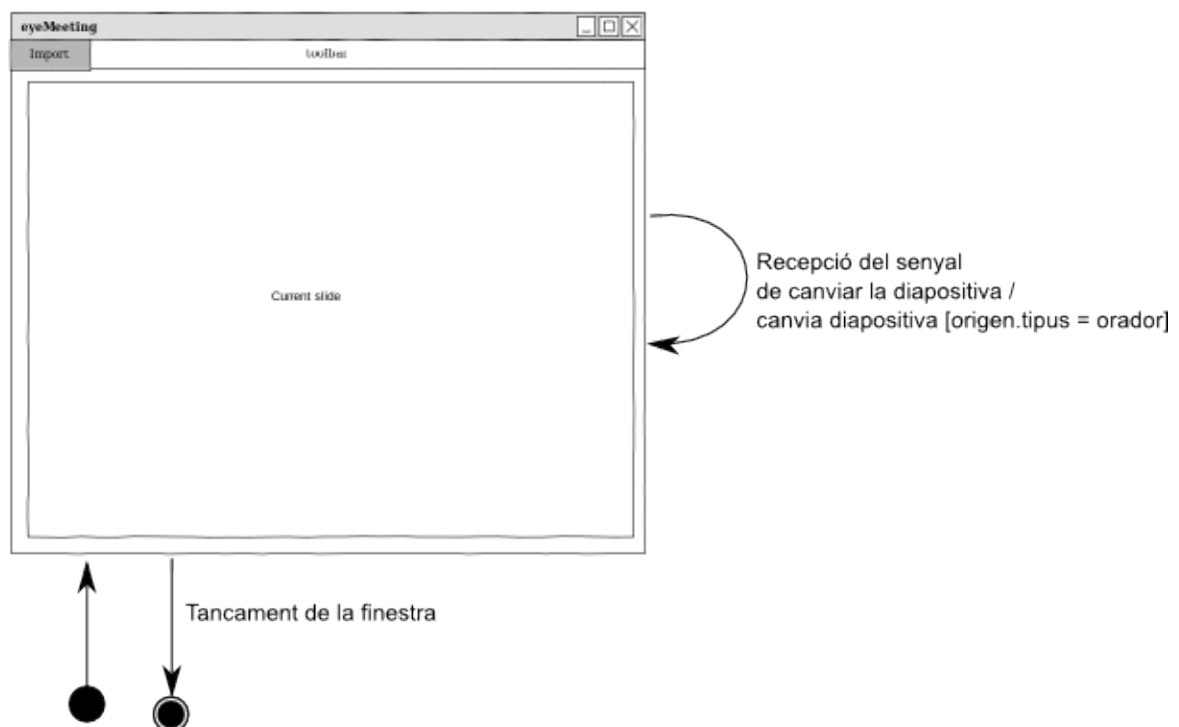
Il·lustració 28: Mapa navegacional de gestionar presentació (1)

Així, el cas d'ús d'iniciar o acabar la presentació té un canvi de vista pels oients: es permet navegar per la presentació lliurement fins que l'orador decideix

començar (Il·lustració 29). De la mateixa manera, el cas d'ús de mostrar diapositiva és també diferent per un oient, com mostra la Il·lustració 30.

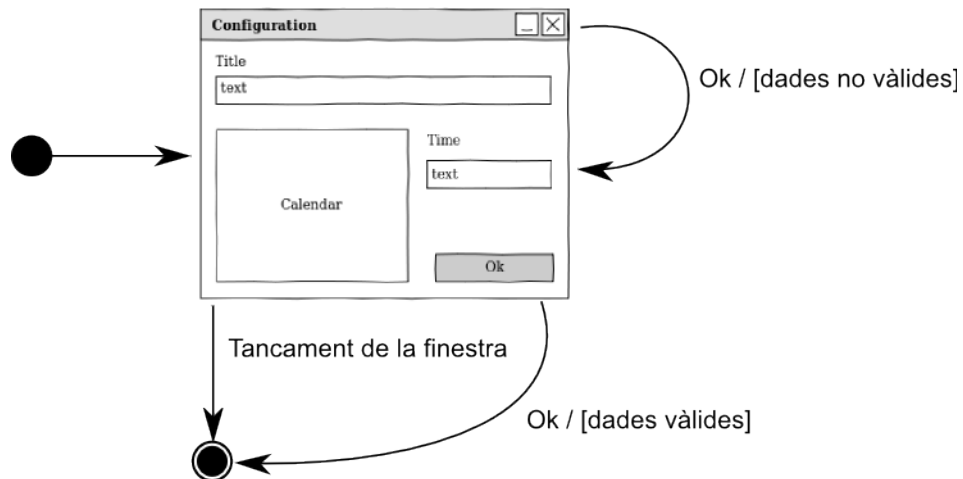


Il·lustració 29: Mapa navegacional de gestionar presentació (2)

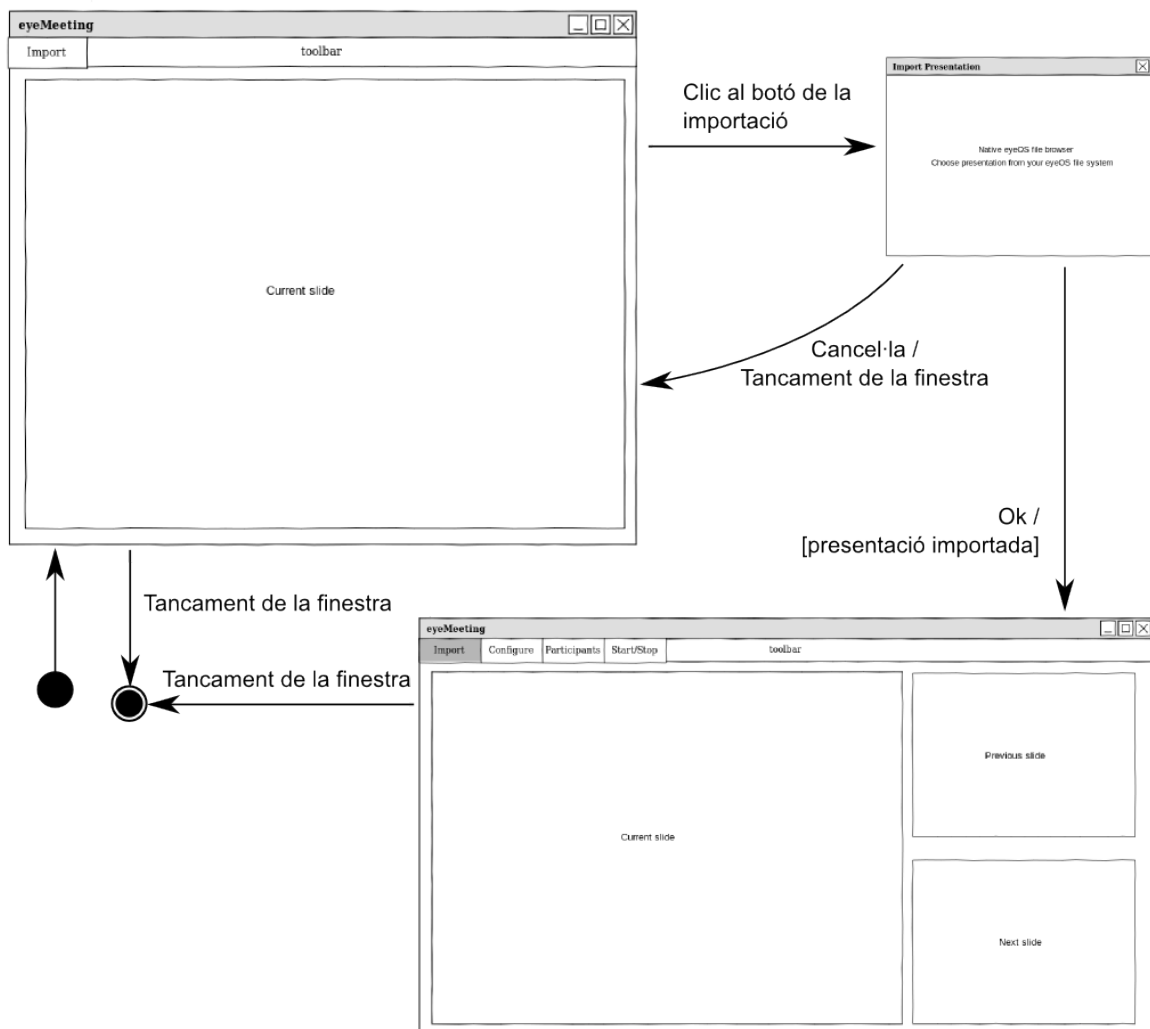


Il·lustració 30: Mapa navegacional de gestionar presentació (3)

El mapa pel cas d'ús d'actualitzar la presentació, per l'orador, en la Il·lustració 31. El cas d'ús d'importar la presentació (orador) és en la Il·lustració 32:

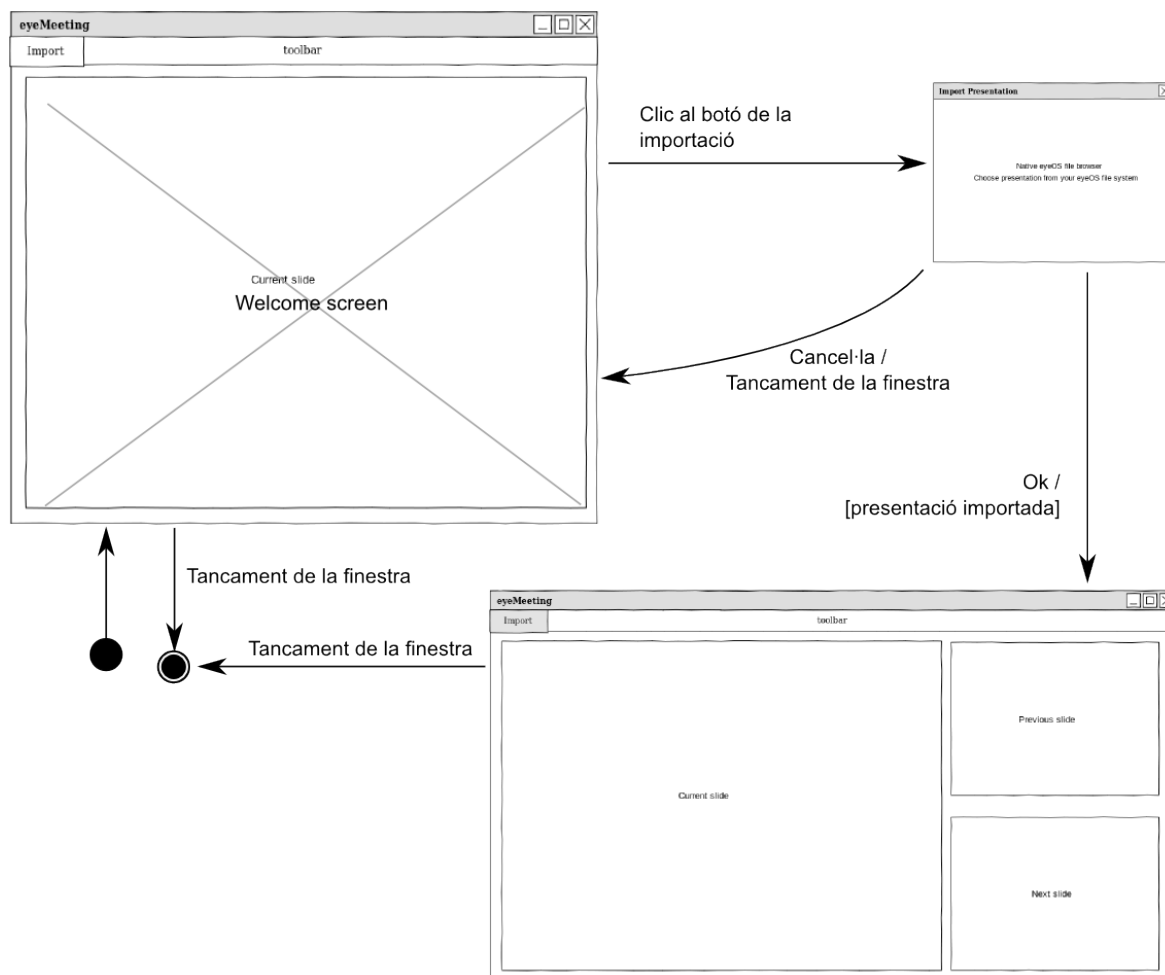


Il·lustració 31: Mapa navegacional de gestionar presentació (4)



Il·lustració 32: Mapa navegacional de gestionar presentació (5)

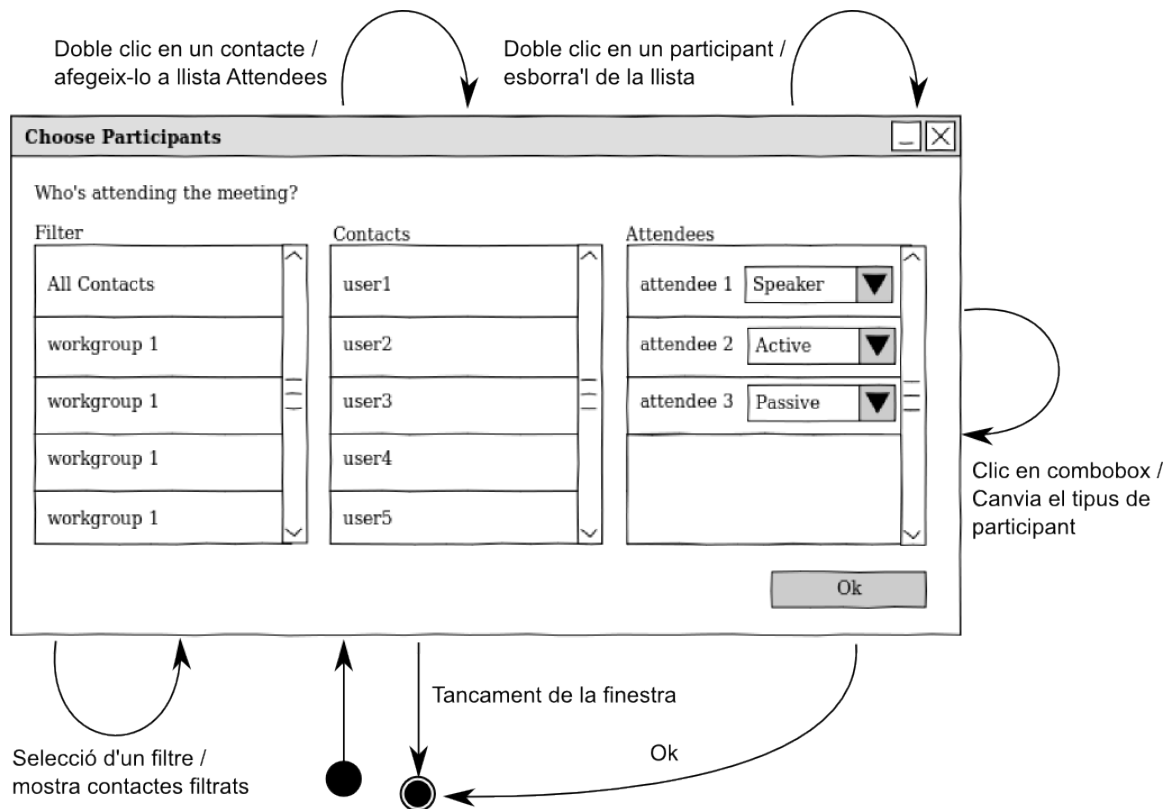
La vista pel cas d'ús importar presentació, si l'executa un oient, és molt similar a l'anterior però a la pantalla principal no es mostren els controls de configuració i gestió de participants.



Il·lustració 33: Mapa navegacional de gestionar presentació (6)

Casos d'ús de gestionar participants

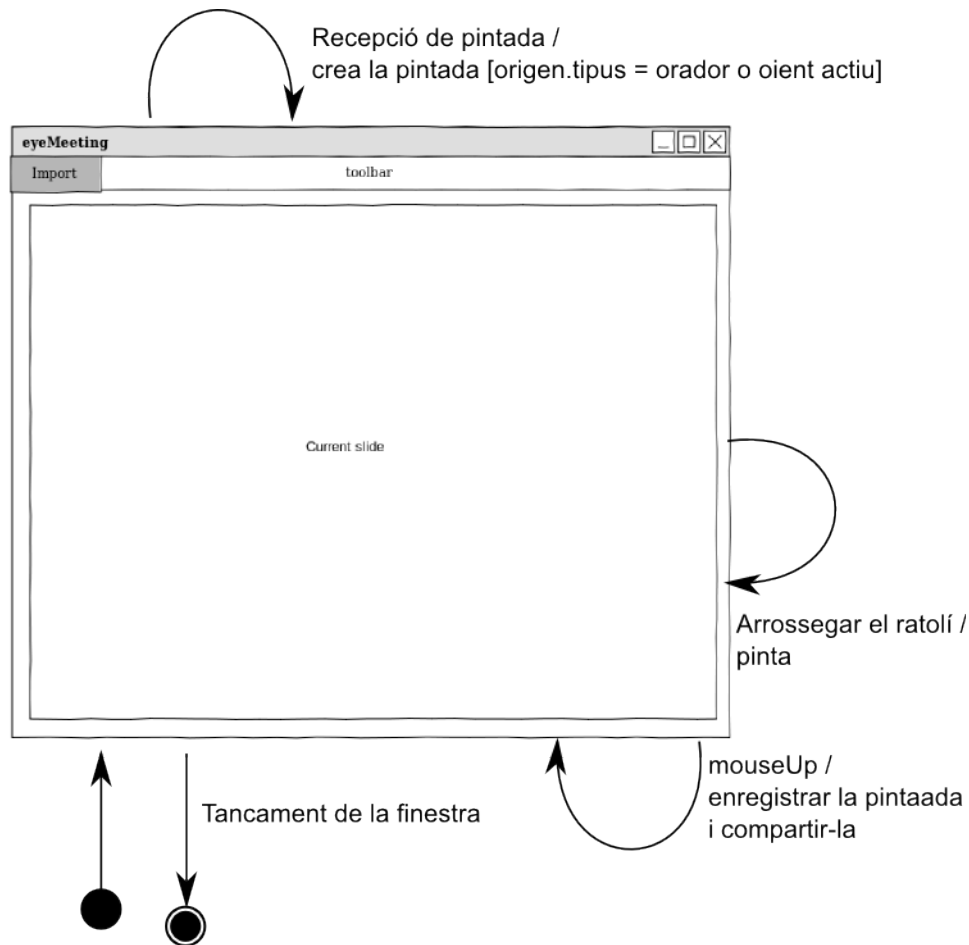
La creació, actualització, obtenció i esborrat de participants es poden mostrar tots en el mateix mapa navegacional de la Il·lustració 34.



Il·lustració 34: Mapa navegacional de gestionar participants

Casos d'ús de gestionar pintades

Crear i obtenir pintades també es poden agrupar en un sol mapa navegacional.



Il·lustració 35: Mapa navegacional de gestionar pintades

Disseny intern

Visió general

El disseny intern defineix com es fa la recepció d'esdeveniments i la comunicació amb capes inferiors. Existeixen 3 classes i un objecte que inicia l'aplicació.

L'objecte és una funció de dues línies obligatòria per les aplicacions d'eyeOS, amb aquest aspecte:

```
1 function eyemeeting_application(checknum, pid, args) {  
2   var app = new eyeos.application.eyemeeting(checknum, pid);  
3   app.drawGUI();  
4 }
```

És el primer que es crida des de l'escriptori i el que marca quin mètode de l'aplicació és el punt d'entrada.

Les altres tres classes corresponen a les tres finestres de l'aplicació: la principal (eyeos.application.eyemeeting), la de configuració (eyeos.eyemeeting.configuration) i la de gestió de participants (eyeos.eyemeeting.participantsChooser). Es comuniquen amb l'eyeos.eyemeeting.presentationManager, que fa les funcions de controlador de domini en el client. Més endavant s'explica en profunditat la seva responsabilitat i les desviacions respecte un disseny ideal.

S'han obviat les classes pròpies de la biblioteca gràfica. No té sentit replicar aquí el manual de Qooxdoo i entrar en un nivell de detall tan gran, si bé és cert que els objectes com les finestres, botons, llistes... formen part de la descripció lògica de la capa de presentació.

Cal tenir present que alguns noms dels fitxers i classes han de seguir una nomenclatura de l'eyeOS o poden no funcionar.

Classes de Qooxdoo

Qooxdoo permet definir classes com es fa amb l'orientació a objectes, tot ignorant moltes de les particularitats de JavaScript. Permet herència (però no múltiple), polimorfisme, mètodes estàtics, membres d'àmbit de classe, constructores i destructores, interfícies, classes abstractes, *singletons* ja fets i *mixins*, un concepte exclusiu del framework.

Una classe de Qooxdoo acostuma a tenir, a diferència de la programació tradicional amb objectes, tres seccions com a mínim: constructora, propietats i membres. En el cas de la interfície gràfica només hi ha les dues primeres.

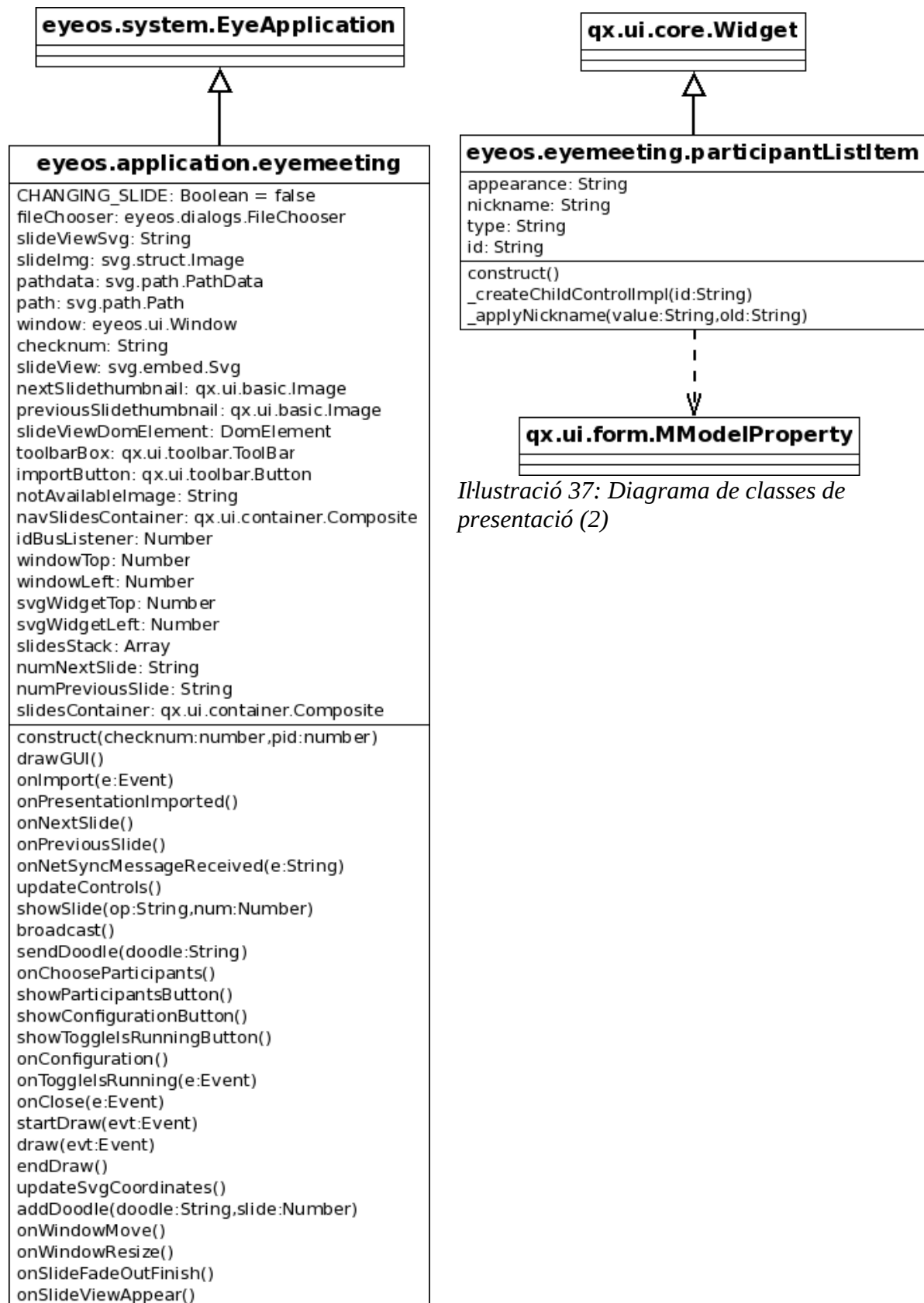
- *Constructor*
Crea una instància de la classe si es crida amb l'operador *new*.
- *Members*
Atributs i mètodes que defineixen la classe. La visibilitat es gestiona amb el conveni següent però sense comprovacions internes:
 - *mètode()* o variable públics
 - *_mètode()* o *_variable* protegits
 - *__mètode()* o *__variable* privatsVe a ser com tenir variables visibles per tota la classe.
- *Properties*

Els atributs tradicionals que necessiten operacions de consulta i modificació (*getters*, *setters*) haurien de ser declarats en aquesta secció i són els únics que haurien de ser exposats fora de la classe. Els *getters* i *setters*, comprovació de tipus i inicialització de valors estan disponibles automàticament. No cal declarar-los.

Diagrames i descripció

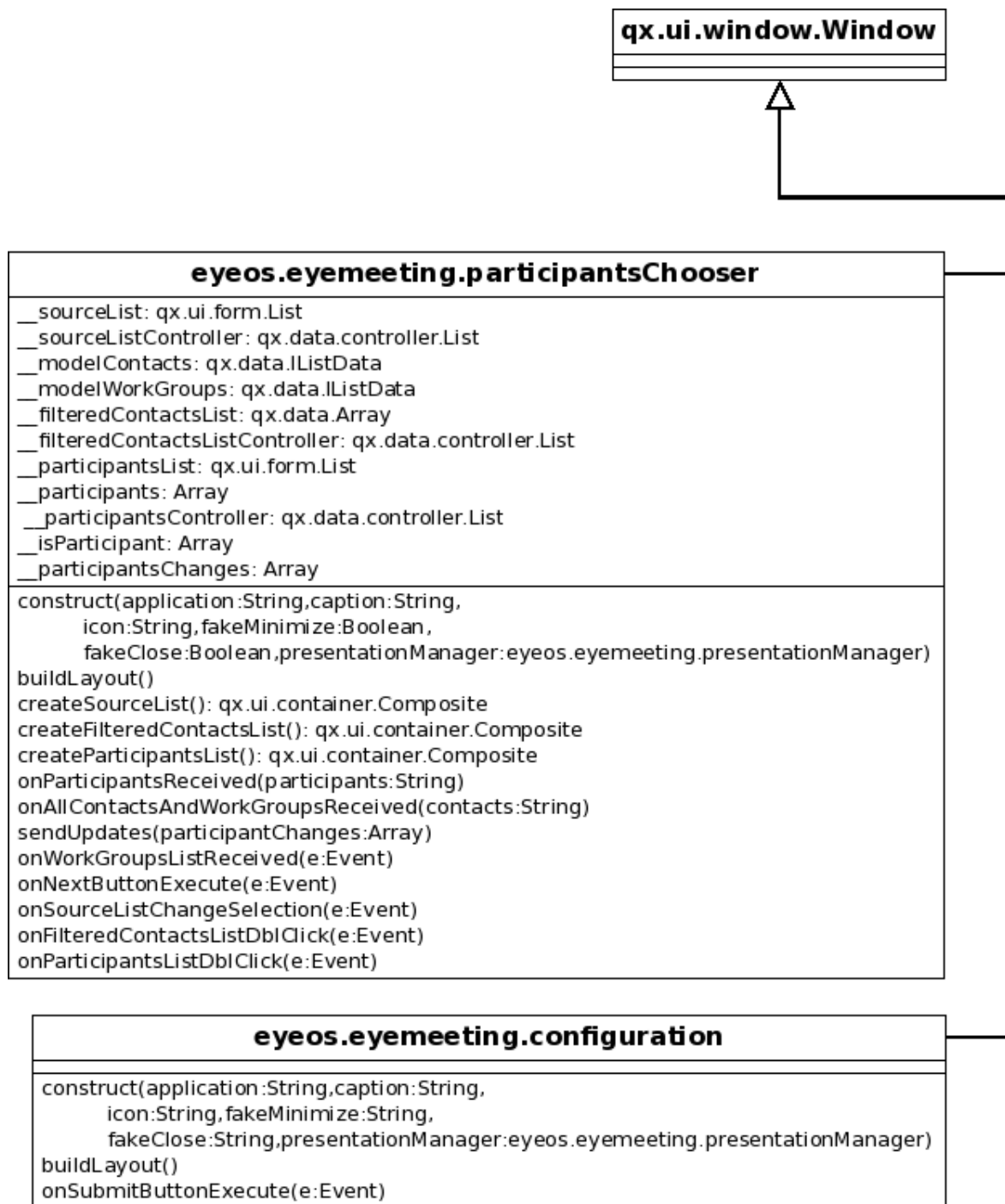
En les il·lustracions 36, 38 i 37 es pot veure la seva definició. En els atributs no es veuen tots els *widgets* perquè en ocasions s'han pogut amagar com variables locals.

Alguns dels mètodes han estat implementats com funcions anònimes per conveniència durant la construcció del programa.



Il·lustració 37: Diagrama de classes de presentació (2)

Il·lustració 36: Diagrama de classes de presentació (1)



Il·lustració 38: Diagrama de classes de presentació (3)

El compliment d'alguns dels requeriments més importants, com per exemple les pintades, és responsabilitat exclusiva de la capa de presentació. La seva lògica és força pesada però en algunes operacions és rutinària. En aquests darrers casos s'ha decidit no incloure el diagrama de seqüència perquè no aporta cap informació rellevant.

També per facilitar la lectura, algunes línies de vida dels diagrames de seqüència no arriben fins el límit inferior de la imatge. Ha calgut fer-ho així per poder oferir el diagrama sencer d'una manera estètica.

Eyemeeting

DrawGUI

Crea la finestra, la barra d'eines, etc. És una acció de poca importància i ho resol el framework.

onNextSlide i onPreviousSlide

Deleguen en un altre mètode el canvi de diapositiva.

updateControls

Controla si les miniatures de diapositives que s'utilitzen per navegar han d'estar habilitades o no segons els límits de la presentació.

onChooseParticipants i onConfiguration

Obren la finestra de gestionar participants i la d'actualitzar la presentació, respectivament.

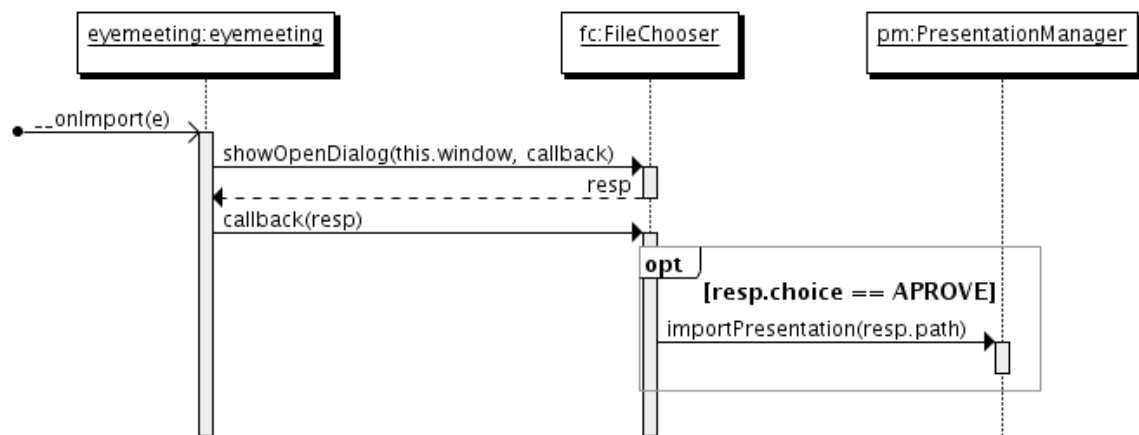
showParticipantsButton, showConfigurationButton, showToggleIsRunningButton

Mostren o no els corresponents controls segons els permisos de l'usuari.

onToggleIsRunning

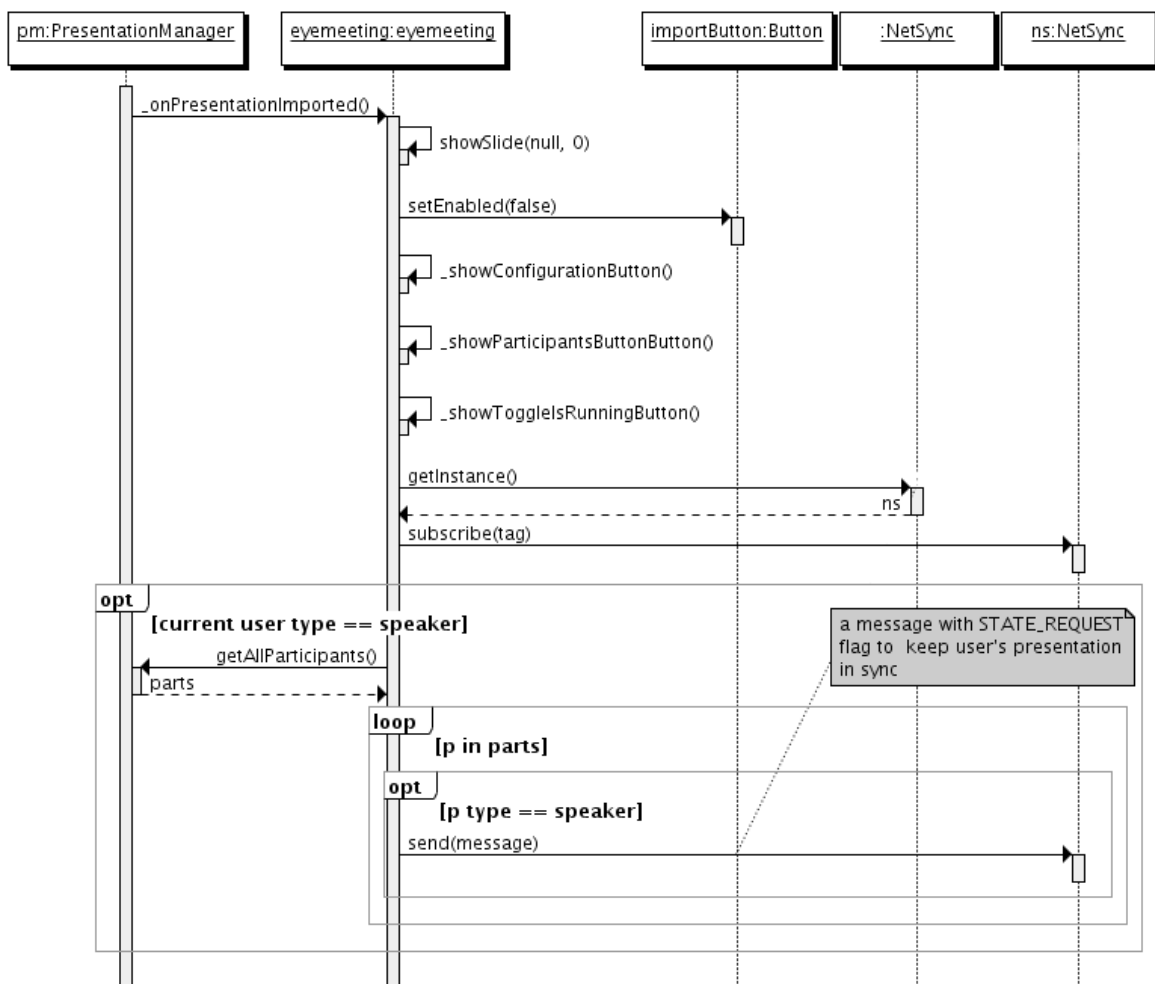
Similar a broadcast: actualitza l'etiqueta del botó toggleIsRunningButton i envia el missatge d'iniciar o acabar la presentació.

onImport



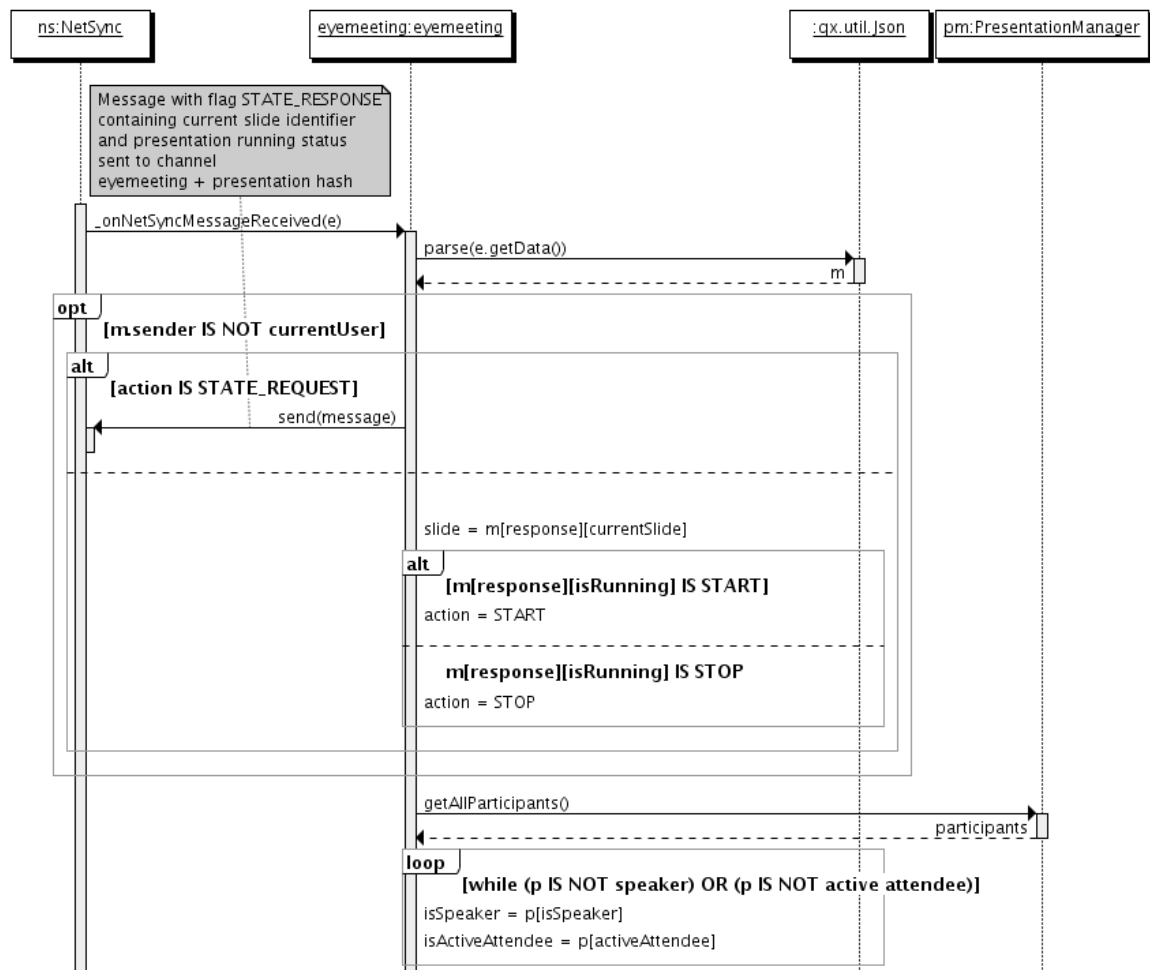
Il·lustració 39: Diagrama de seqüència: `onImport`

onPresentationImported

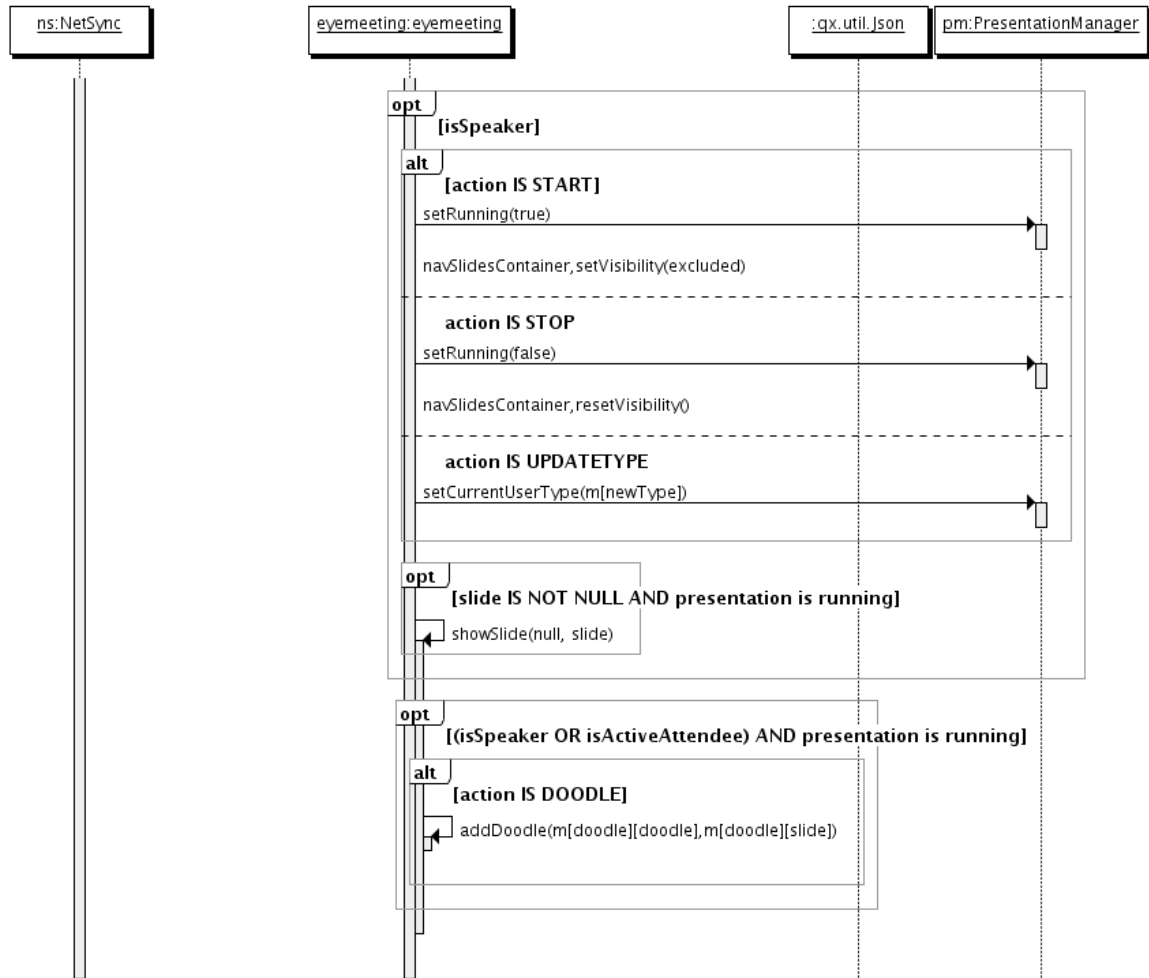


Il·lustració 40: Diagrama de seqüència: `onPresentationImported`

onNetSyncMessageReceived



Il·lustració 41: Diagrama de seqüència `onNetSyncMessageReceived` (1)



Il·lustració 42: Diagrama de seqüència onNetSyncMessageReceived (2, continuació)

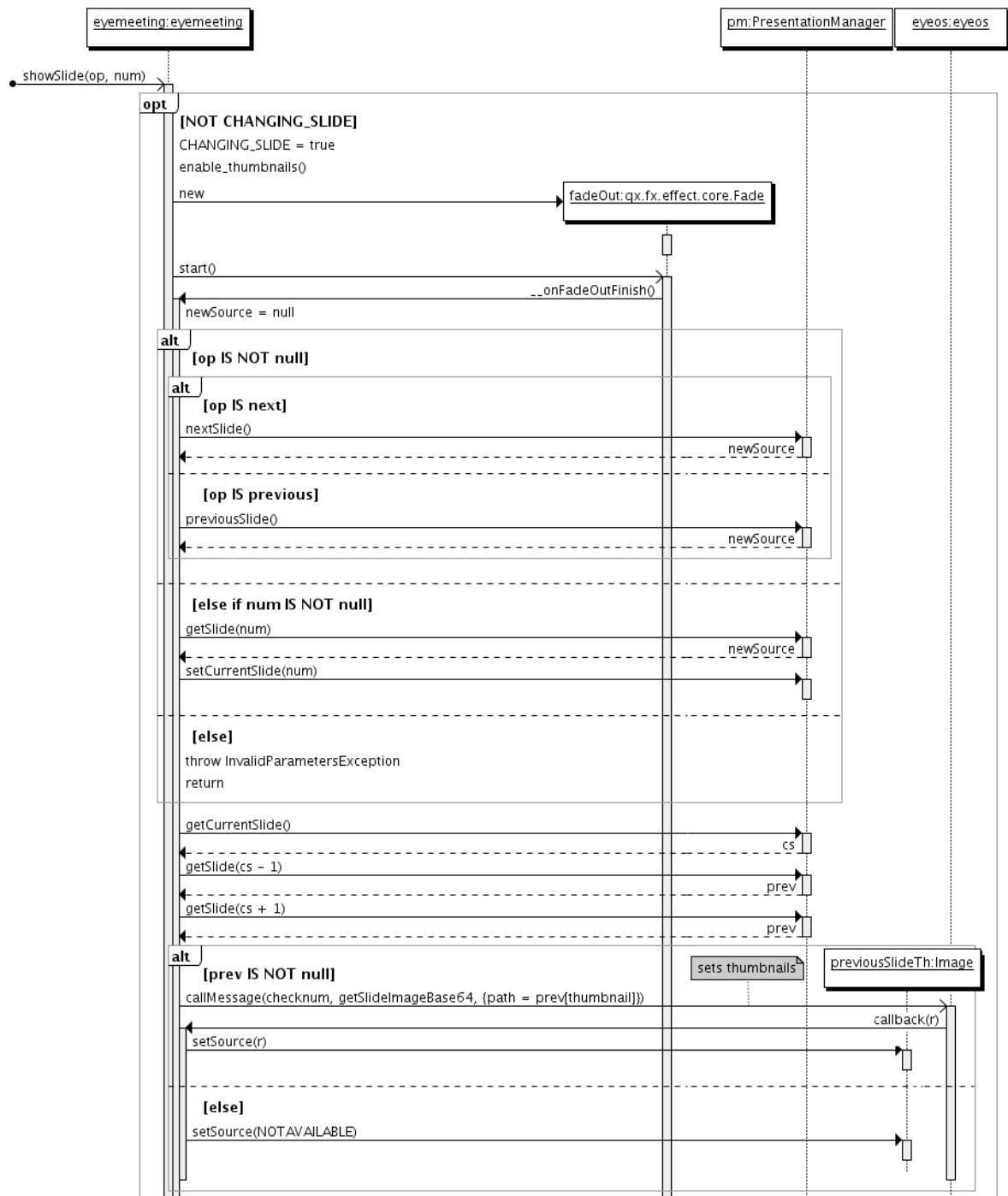
showSlide

Aquesta operació és una de les més importants en el programa. El diagrama s'ha dividit en 3 parts per permetre'n la lectura. A grans trets:

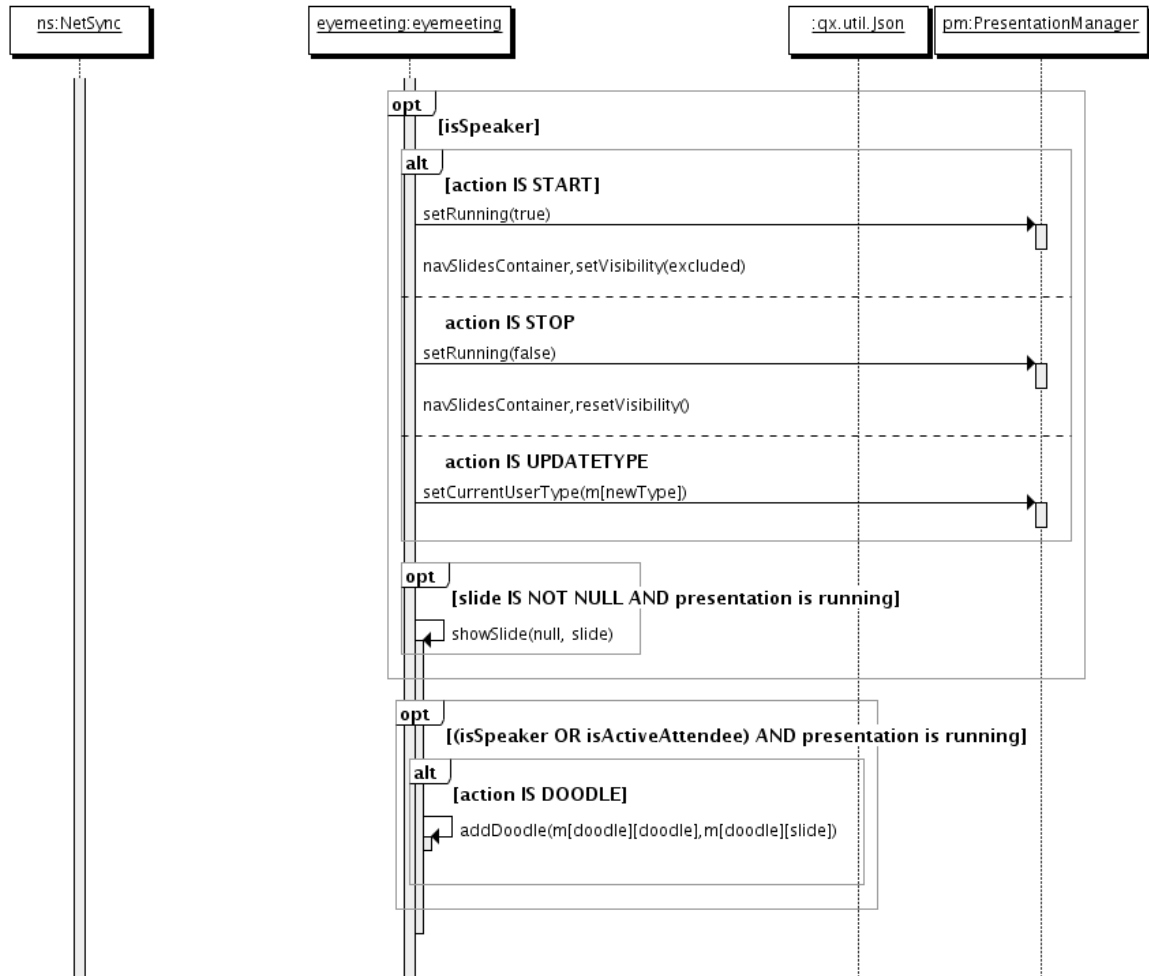
1. Posa el semàfor CHANGING_SLIDE a false
2. Desapareix la diapositiva actual
3. Actualitza les miniatures
4. Actualitza la diapositiva actual i la fa aparèixer
5. Envia el senyal als participants.

En el capítol 6.2.3 Implementació - SVG s'explica en detall com és el component que encapsula la diapositiva actual.

La primera part, en la Il·lustració 43:

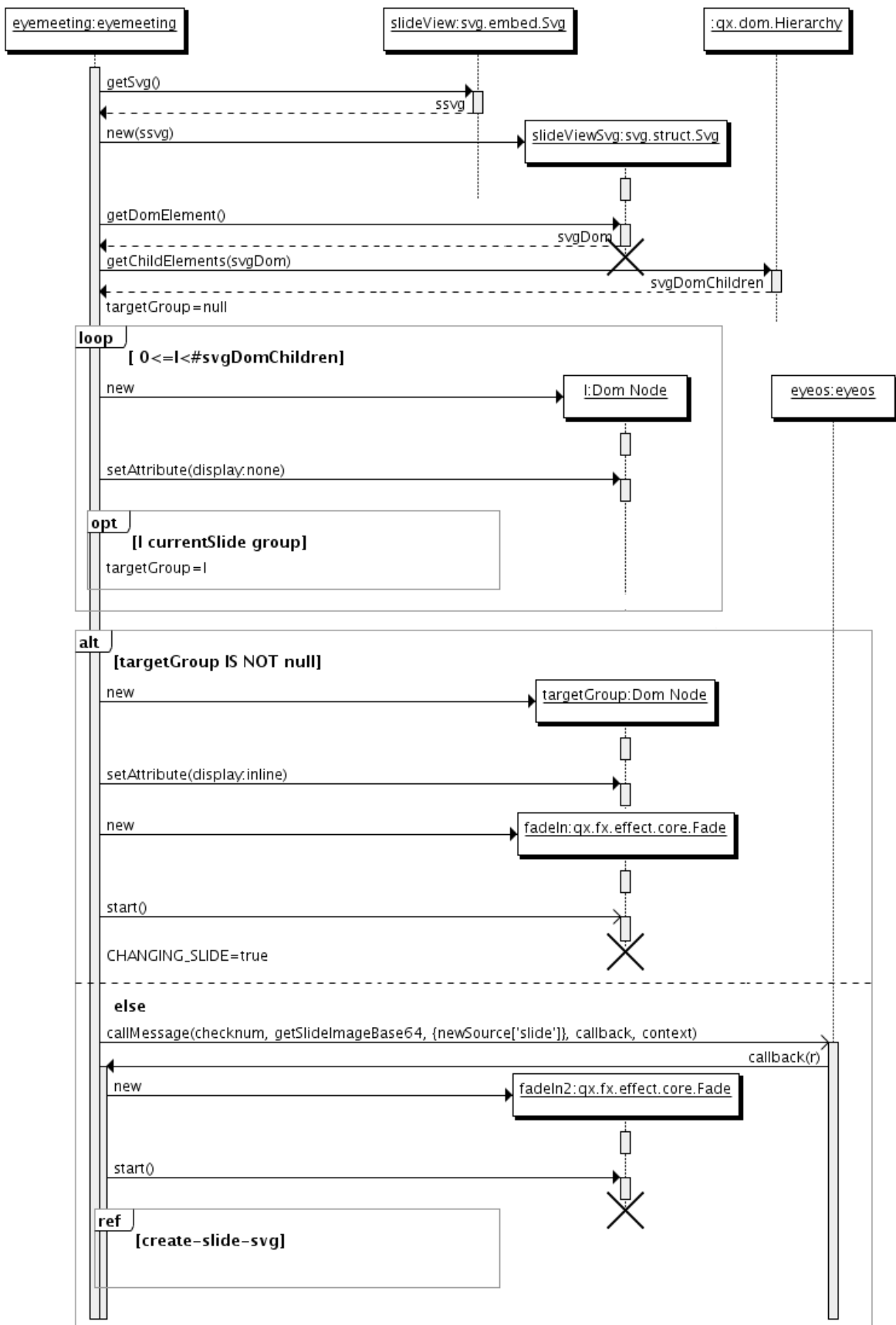


Il·lustració 43: Diagrama de seqüència de showSlide (1A)



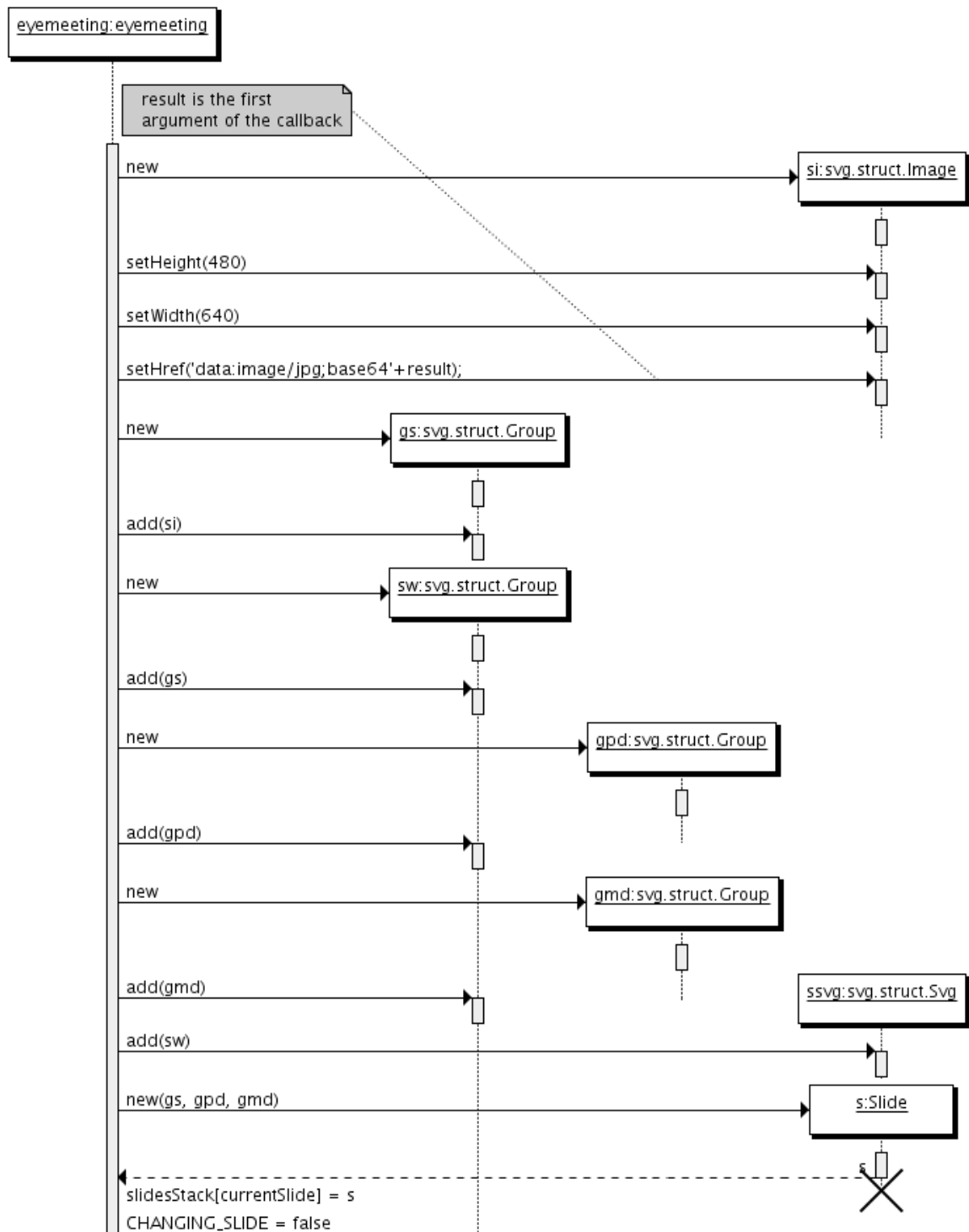
Il·lustració 44: Diagrama de seqüència de showSlide (1B)

La segona part, sobre SVG, en la Il·lustració 45



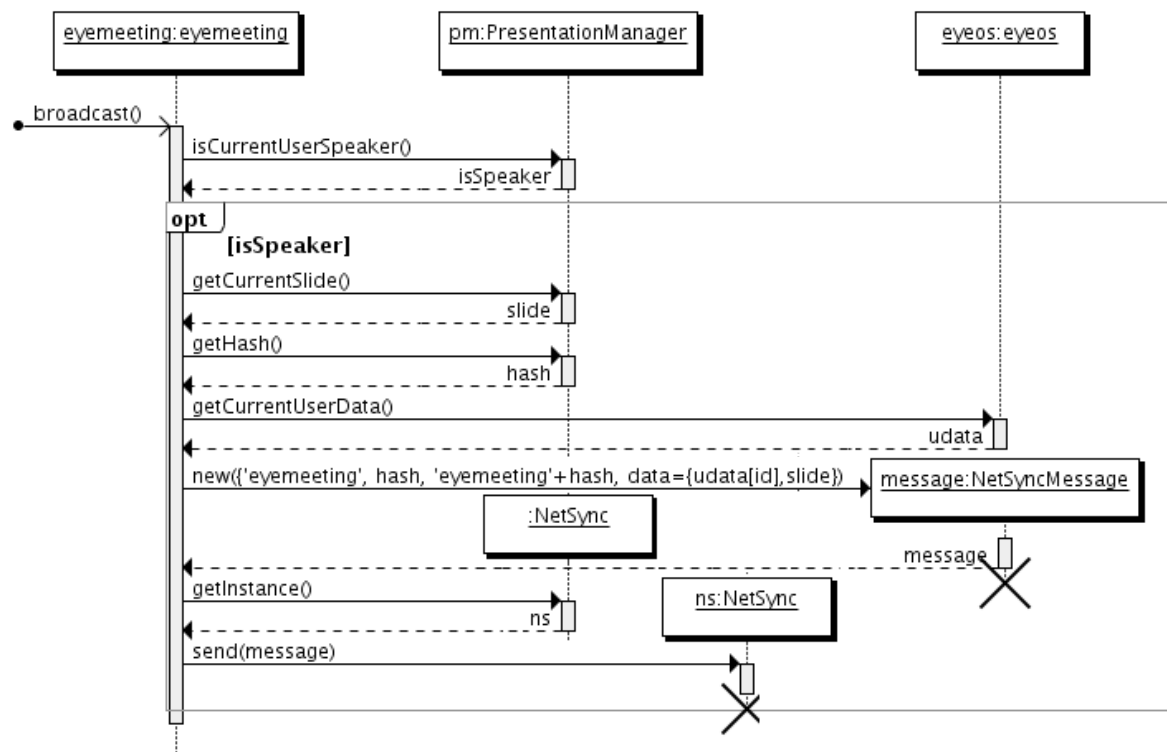
Il·lustració 45: Diagrama de seqüència de `showSlide` (2)

i la tercera part, que correspon al darrer marc de la Il·lustració 45 amb referència *create-slide-svg*, en la Il·lustració 46:



Il·lustració 46: Diagrama de seqüència de *showSlide (3) -- create-slide-svg*

broadcast



Il·lustració 47: Diagrama de seqüència de broadcast

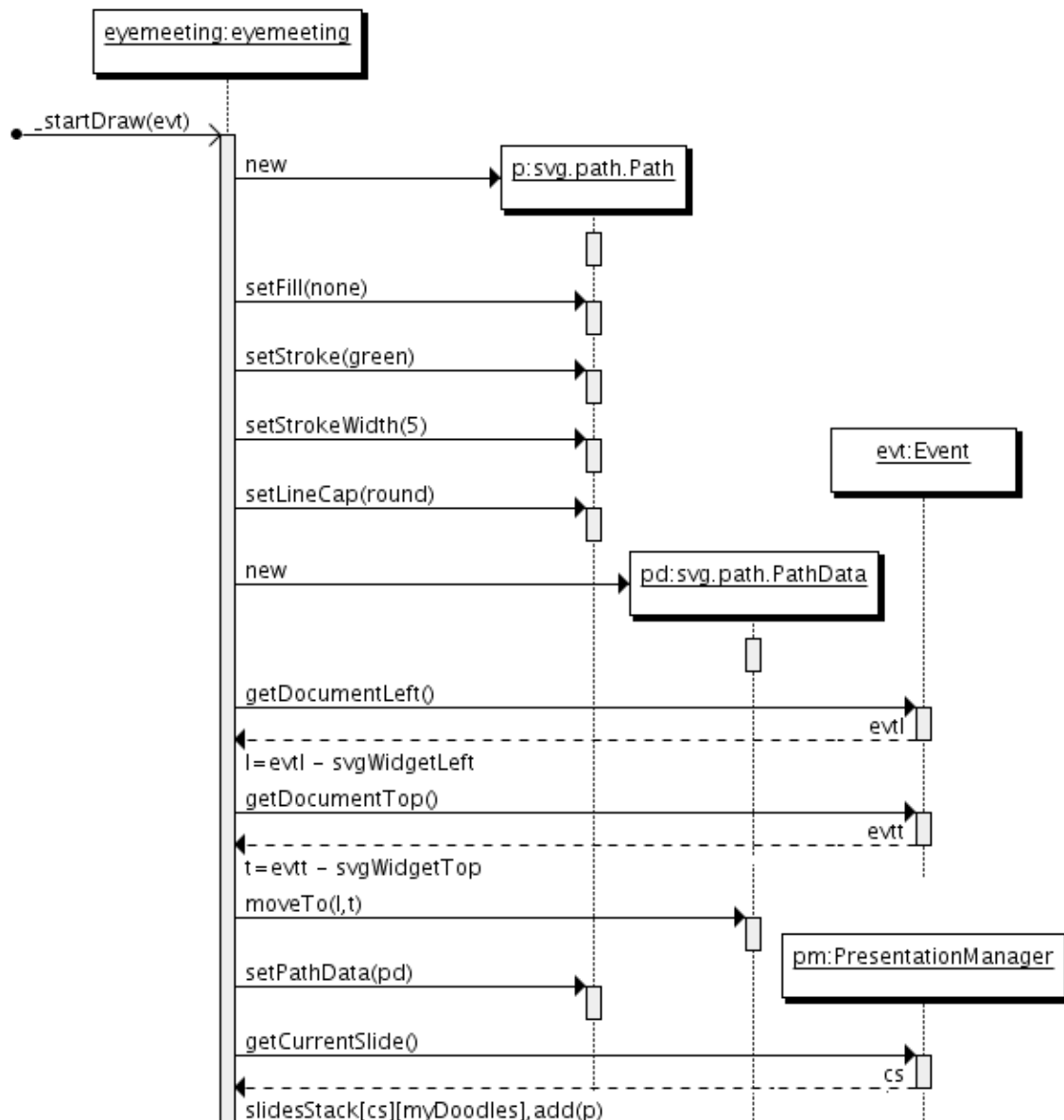
sendDoodle

Similar a l'anterior

onClose

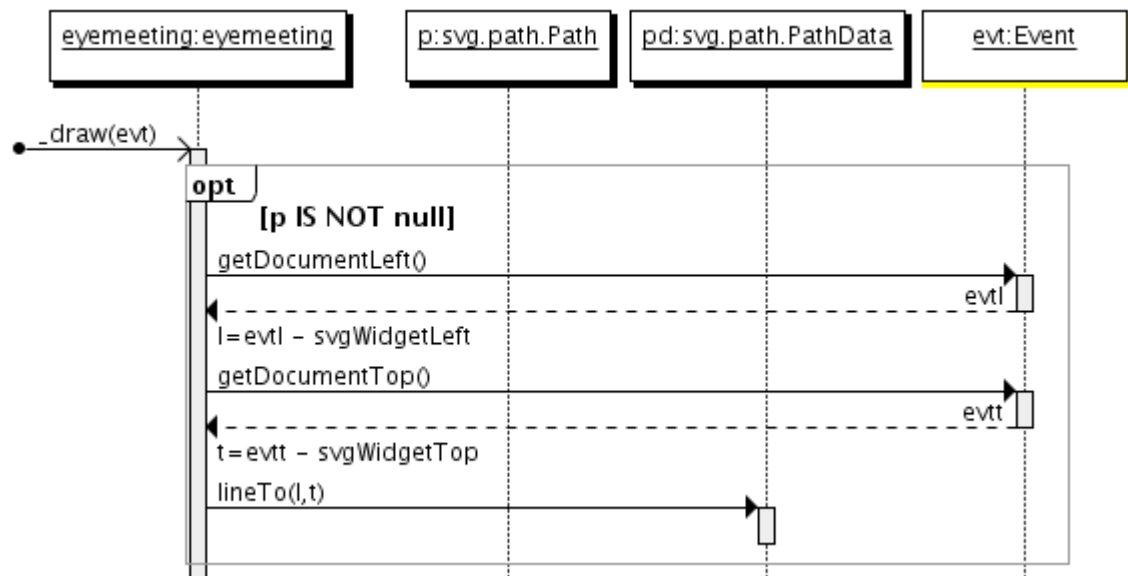
Similar a broadcast. Abans de tancar s'envia senyal d'acabar la presentació al canal, es dóna de baixa del canal i s'esborra el listener del d-bus.

startDraw



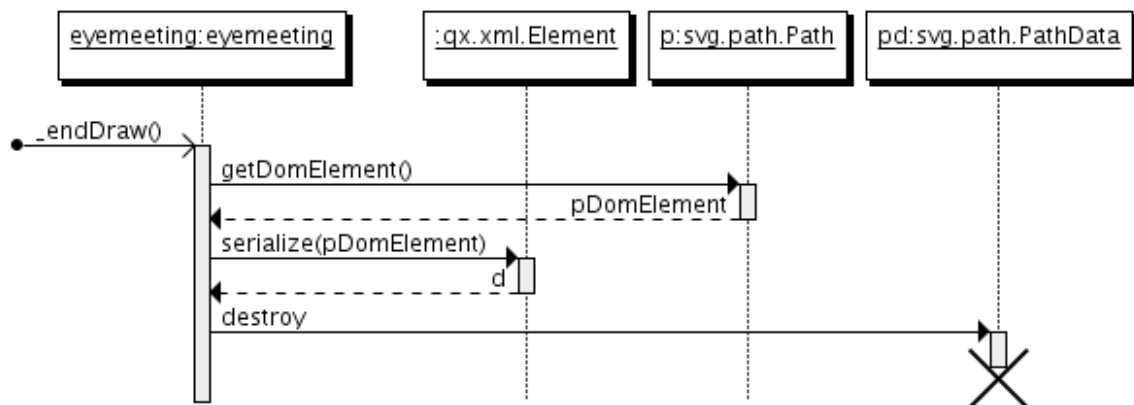
Il·lustració 48: Diagrama de seqüència de `startDraw`

draw



Il·lustració 49: Diagrama de seqüència de draw

endDraw

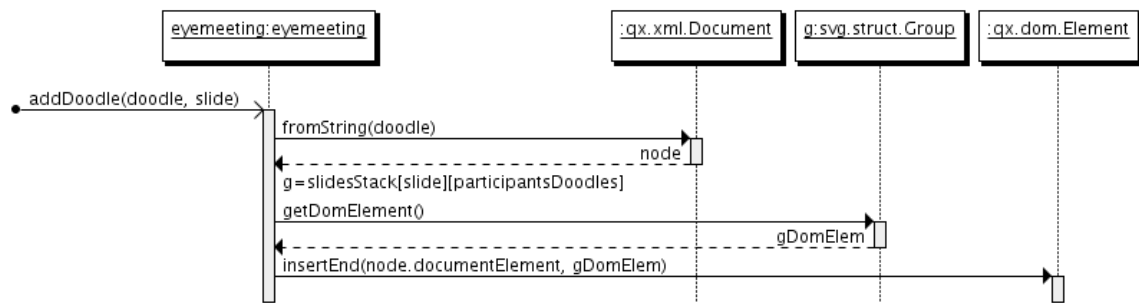


Il·lustració 50: Diagrama de seqüència d' endDraw

updateSvgCoordinates

Operació trivial que actualitza variables membre amb les coordenades del llenç svg per qüestions de rendiment.

addDoodle



Il·lustració 51: Diagrama de seqüència d' `addDoodle`

Configuration

buildLayout

Operació igual que `drawGUI` i altres `_buildLayout`. Crea la finestra i hi situa els controls.

onSubmitButtonExecute

Operació trivial que comprova que les dades siguin vàlides i compleixin les restriccions del model. Si és així, actualitza el model i ho notifica als participants.

ParticipantsChooser

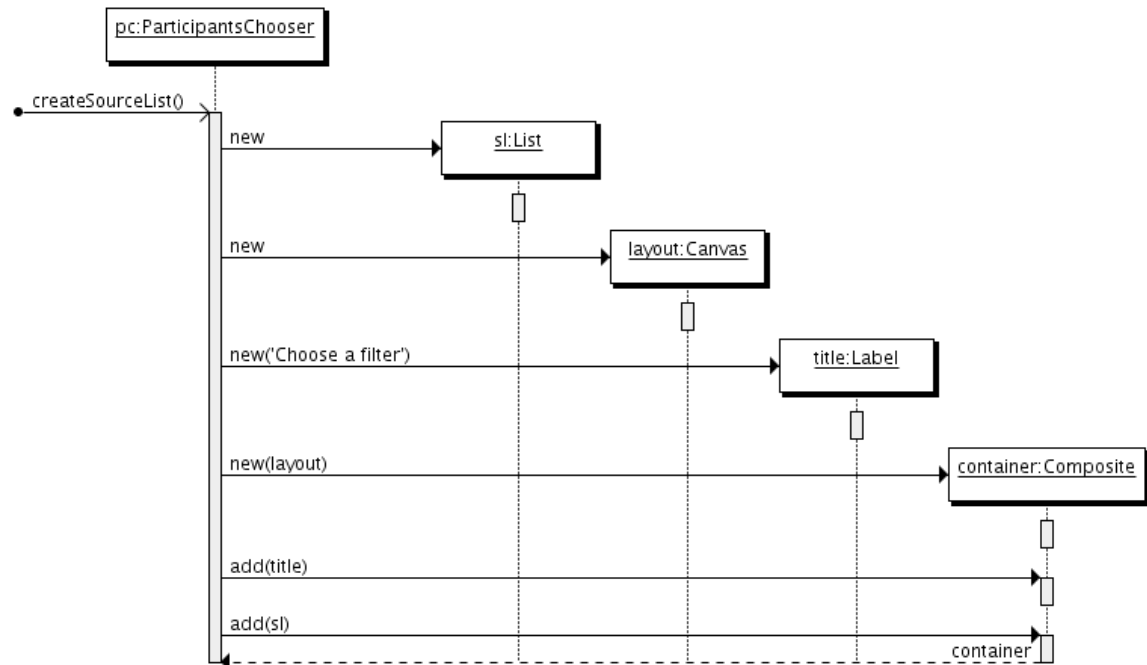
Aquesta classe conté, entre altres controls, tres llistes similars: `workgroups`, `contactes` i `participants`. La llista dels `workgroups` serveix per filtrar els contactes. La llista de participants mostra els usuaris que han estat convidats. Les tres funcionen d'una manera molt similar, amb model-vista-controlador, i no té sentit fer els diagrames per cadascuna. En la documentació de Qooxdoo a la Bibliografia i referències es pot consultar el funcionament de les llistes en profunditat.

buildLayout

Operació trivial que col·loca els controls a la finestra i demana al servidor les dades necessàries per crear els models de les 3 llistes. Com que tot és asíncron, s'han definit operacions per atendre les *callbacks* quan el sistema indiqui que ja s'han rebut les respostes del servidor.

createSourceList, createFilteredContactsList i createParticipantsList

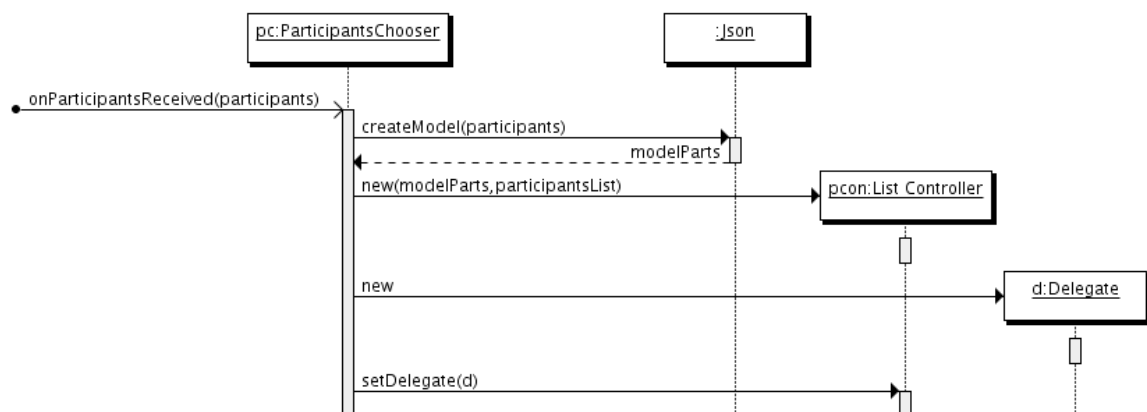
Retornen un component que es pot incloure a la finestra. Encapsula la construcció de la llista i dels seus elements a partir del model que es crea amb la resposta del servidor. Les tres són pràcticament idèntiques:



Il·lustració 52: Diagrama de seqüència de createSourceList

onParticipantsReceived, onAllContactsAndWorkGroupsReceived

De tres llistes només calen dades del servidor per dues. Aquestes operacions són les dues callbacks necessàries. La lògica ha de ser similar però es distingeixen en el Delegate que utilitzen.

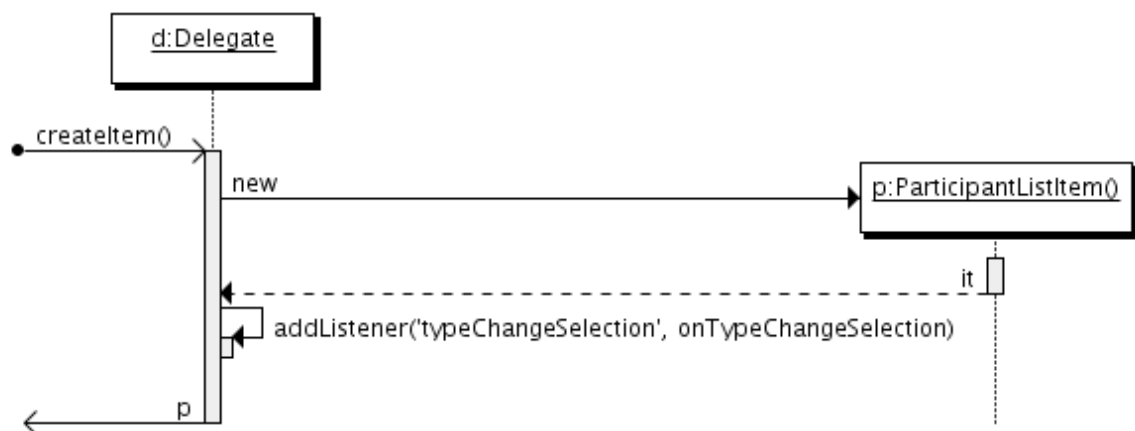


Il·lustració 53: Diagrama de seqüència d' onParticipantsReceived

Una llista (`qx.ui.form.List`) es fa habitualment amb la classe `qx.ui.form.ListItem`. La llista de participants ha necessitat estendre-la

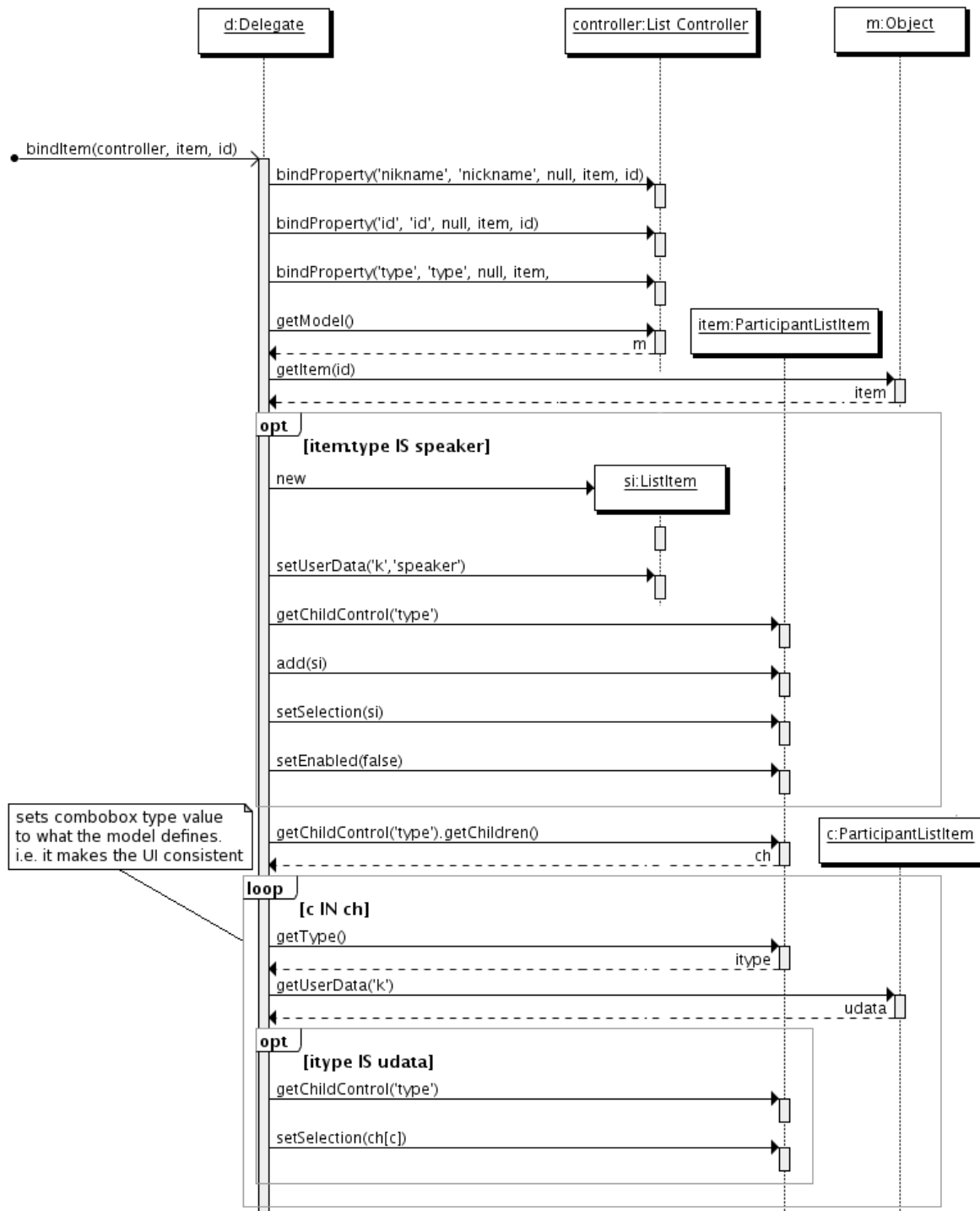
(eyeos.eyemeeting.ParticipantListItem) per poder tenir un combobox en cada element i fa la lògica lleugerament més complexa. Aquesta característica es fa redefinint operacions del Delegate, que s'assigna quasi sempre al controlador (qx.data.controller.List). És habitual redefinir els mètodes:

- *createItem* que es crida en crear cada element de la llista
- *bindItem* en què es defineix a quin subcomponent de l'element de la llista s'enllaça cada propietat del model.
- *ConfigureItem* en què s'hauria de definir com és el contingut de l'ítem.
- *Filter* en què es determina quins elements del model formen part de la llista.



Il·lustració 54: Diagrama de seqüència de *createItem*

El Delegate de la llista de participants amb les seves operacions sobreescrites:



Il·lustració 55: Diagrama de seqüència de bindItem

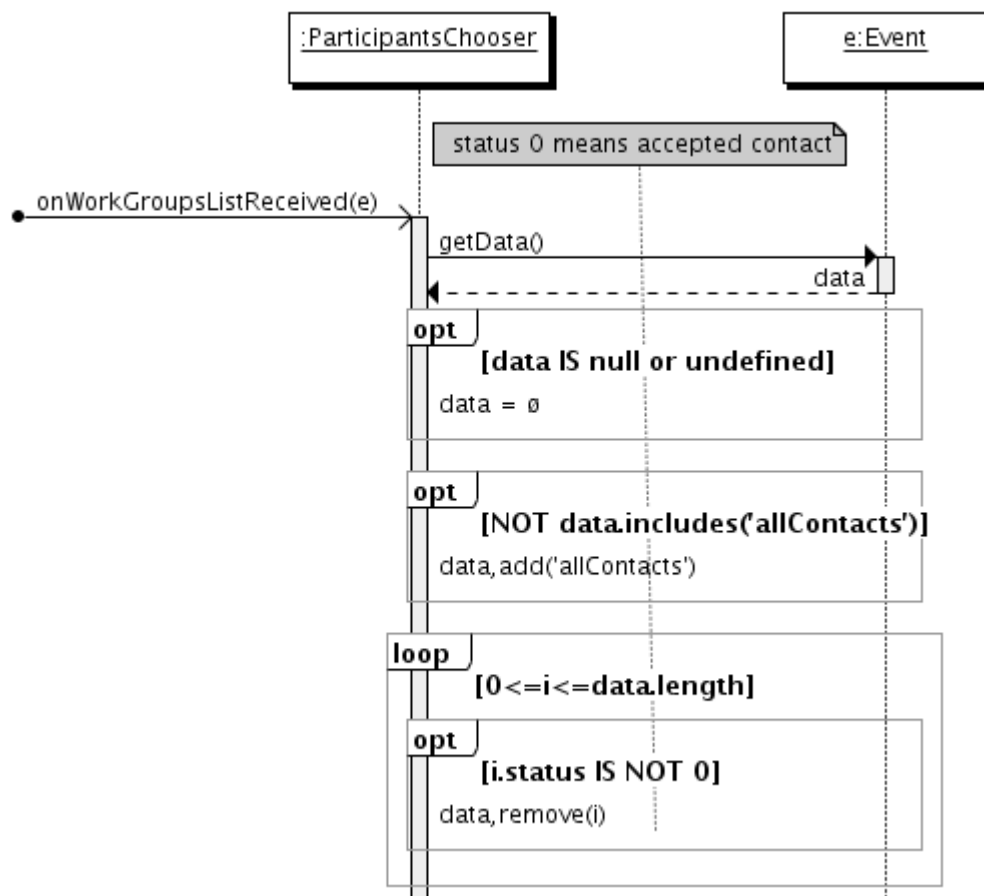
D'acord amb la documentació de Qooxdoo, part d'aquest codi (a partir del darrer bindProperty) hauria de ser en un altre mètode configureItem, però sembla que hi ha un error en la biblioteca i no s'ha pogut fer. S'ha estimat convenient posar-lo en bindItem per l'ordre observat en les crides a aquests mètodes.

El delegate de la llista inicial de contactes és simplement una funció filtre per no incloure-hi els participants i per triar els contactes que pertanyin al workgroup seleccionat en la primera llista.

sendUpdates

Com altres mètodes que utilitzen NetSync (p. ex. eyemeeting::broadcast) , sendUpdates envia les notificacions als participants indicats en el seu paràmetre.

onworkGroupsListReceived



Il·lustració 56: Diagrama de seqüència de `onWorkGroupsListReceived`

onnextButtonExecute

Operació trivial que delega a `presentationManager::updateParticipants` i envia les notificacions als destinataris adjacents en el `callback`.

onSourceListChangeSelection

Operació trivial que reassigna el delegate al controlador de la llista filteredContactsList amb un filtre nou que permeti triar només els contactes que pertanyin al workgroup seleccionat en la primera llista.

onFilteredContactsListDbClick, onParticipantsListDbClick

Operació trivial que modifica el model de la llista de participants per afegir-hi el contacte en què s'ha fet doble clic. Actualitzen estructures auxiliars per qüestions de rendiment.

ParticipantListItem

Extén la classe Widget per crear un ListItem a mida de les necessitats. Un ListItem estàndard té una icona i una etiqueta de text, però la llista que utilitza ParticipantListItem necessitava una etiqueta de text i un combobox.

_createChildControlImpl

Operació trivial que col·loca els controls en el Widget.

_applyNickname(value, old)

Operació trivial i típica de Qooxdoo. Es crida després de setNickname() i abans d'enregistrar els canvis. En aquest cas, ha de ser-hi per requeriments del framework: necessita obtenir el control del component afectat en el widget amb getChildControl¹⁶.

5.2.2.3 Disseny de la capa de domini

Visió general

La capa de domini és responsable de la implementació de les funcions del sistema. S'assabenta dels esdeveniments que rep, en controla la validesa, canvia l'estat del domini, executa les funcions encomanades, s'assabenta de les consultes que cal fer, n'obté els resultats i comunica les respostes. Coneix com satisfer les peticions de l'usuari, però ignora on es guarden les dades i com es presenten a l'usuari. [ES2]

Mentre que la capa de presentació és tota en el client, el domini està distribuït. En

¹⁶ Consulteu documentació de Qooxdoo indicada a Bibliografia i referències

les següents pàgines s'explica quines classes formen part del domini, quines responsabilitats tenen i com satisfan part dels requeriments. S'ha distribuït la capa, no els seus objectes¹⁷.

Façana remota

Inicialment es va pensar en tenir una façana remota davant el controlador del servidor per oferir respostes de granularitat més gruixuda al client. Aquest patró és útil quan ja es disposa d'una interfície d'objecte de granularitat fina i es vol conservar però aquest nivell de granularitat fa que els accessos remots siguin molt costosos¹⁸.

Es va desestimar perquè es podia dissenyar un controlador adequat als requeriments d'un sistema distribuït sense perjudicar severament la canviabilitat i o el poder donar noms molt descriptius als membres de la classe. S'ha dissenyat amb operacions de granularitat suficientment gruixuda.

L'eyeOS requereix tenir un sol objecte al servidor que atengui les peticions remotes. Amb aquesta restricció s'ha construït `eyemeetingApplication` perquè tingui el paper d'un controlador façana. S'entén que hauria de ser singleton, però l'eyeOS demana que sigui una classe abstracta. Cal ser conscients que a la llarga el controlador pot créixer molt i patir de baixa cohesió, però és una restricció del sistema i no es pot escollir.

DTO

És habitual aplicar el patró *data transfer object* quan es defineix la interfície d'operacions remotes per agrupar crides i fer-les de granularitat més gruixuda. En aquest cas s'ha adaptat a les particularitats de JavaScript: un objecte serialitzat té el mateix aspecte que una matriu associativa de PHP. En comptes de tenir pràcticament una classe DTO per cada operació que en arribar al client tindria aspecte de matriu, s'ha optat per fer servir matrius a tot arreu. PHP no permet declarar tuples i en l'eyeOS és una pràctica habitual retornar matrius associatives al client. Probablement utilitzar objectes hauria estat més segur i hauria permès garantir la consistència entre tipus de dades de retorn però, en el moment de dissenyar el sistema, es van seguir els consells de l'oficina d'eyeOS i va semblar

17 [Fowler] «My First Law of Distributed Object Design: Don't distribute your objects»

18[Fowler] Capítol «Remote Facade» i capítol 5.1 L'entorn: eyeOS7 d'aquest document.

adient fer-ho d'aquesta manera.

PresentationManager

En l'eyeOS és comú trobar classes *Manager. Per exemple, ContactManager. En general, és una classe del domini de l'eyeOS que pot ser utilitzada per diverses aplicacions. Té una sèrie de mètodes que escolten notificacions que puguin arribar pel d-bus de l'eyeOS, mètodes per consultar i modificar els atributs, i els atributs que només s'actualitzen quan es rep notificació de canvis.

Per exemple: la barra superior de l'escriptori permet a l'Alícia consultar tots els contactes, però també es pot fer quan vol convidar contactes a un workgroup des d'una altra vista. Mentre l'Alícia obre el programa d'afegir contactes a un workgroup, res no impedeix a en Bernat (un dels contactes de l'Alícia) esborrar-la. Això podria provocar errors d'integritat perquè l'Alícia intentaria afegir en Bernat, quan el sistema només permet afegir contactes a un workgroup. Per evitar-ho, en Bernat ha enviat un senyal a l'Alícia i ella actualitza (fa una crida remota) els seus contactes. Així manté de manera eficient un domini sempre correcte. Cal fixar-se en dues coses:

- L'àmbit d'aquestes operacions és tot l'eyeOS
- Diversos programes utilitzen aquest domini

Sembla un patró que es pugui aplicar també a l'aplicació d'aquest projecte: una presentació les dades de la qual es consulten des de diversos punts (enviament del senyal de canviar de diapositiva a tots els participants, la gestió dels participants, etc), és possible que es modifiquin des de fora (p. ex. diversos participants) o des de diverses finestres (p. ex. La principal i la de gestió de participants) i s'ha d'evitar demanar-les contínuament al servidor. A primer cop d'ull sembla correcte i quan es va dissenyar l'arquitectura del sistema es van seguir els consells de l'oficina. Crear un PresentationManager va ser una solució raonable.

El cert és que aquesta estratègia no és del tot recomanable per l'eyeMeeting:

- La façana del servidor ja disposaria d'una operació per cada esdeveniment. PresentationManager estaria fent de proxy sense aportar grans beneficis.
- Els objectes del client estan vius fins que es refresca la pàgina. No és imprescindible una memòria cau com la dels contactes.

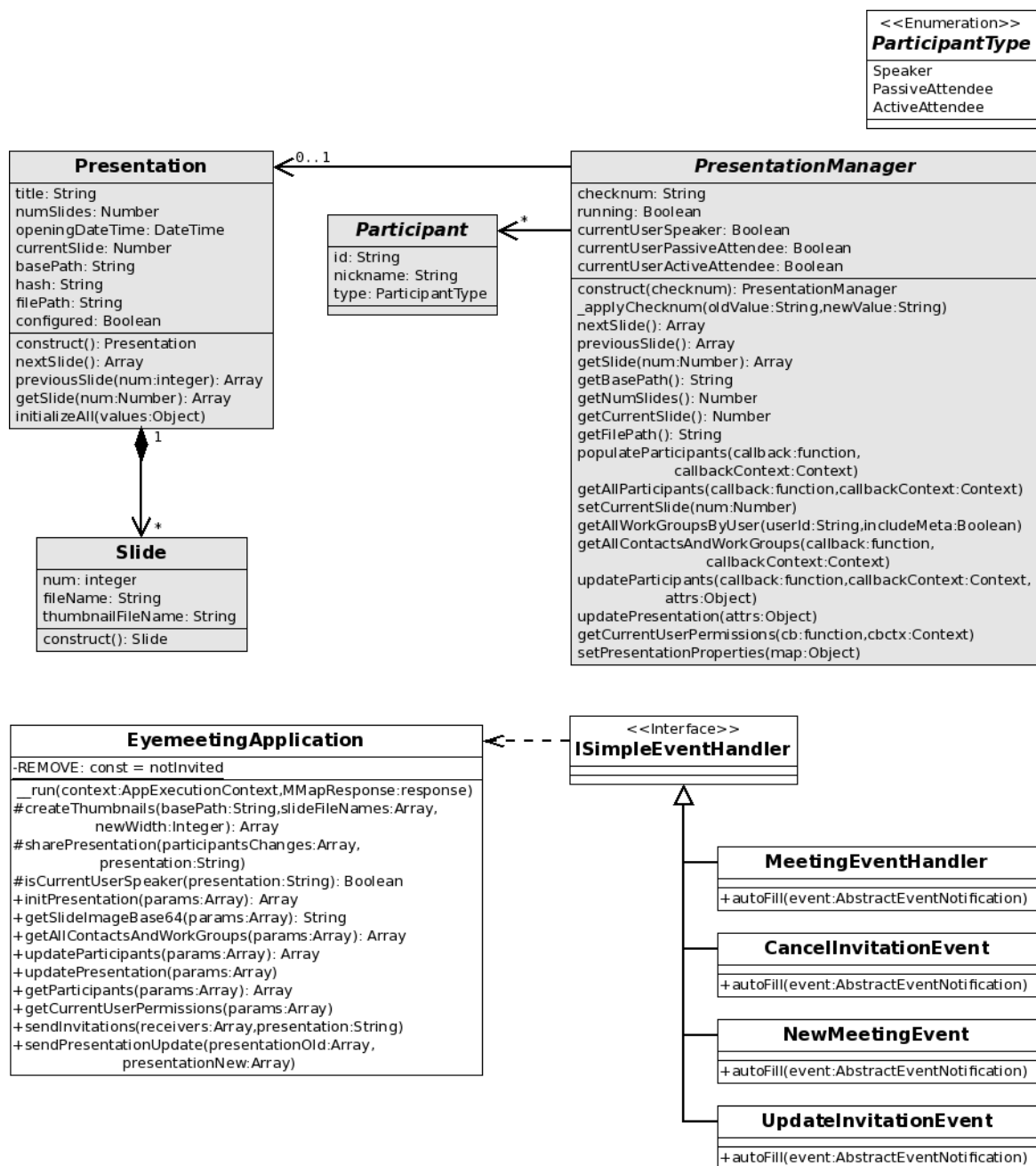
- El nom «PresentationManager» pot portar a confusió amb la classe «Presentation». El Manager és, en el moment de tancar el projecte, un controlador del domini en el client, mentre que Presentation és una representació d'una presentació (del conjunt de diapositives però no de l'acció de presentar).
- No pot ser un *singleton*. Aquest patró permet tenir una sola instància d'una classe en tot l'eyeOS, però es necessita que sigui única en l'aplicació. Fer PresentationManager *singleton* provoca errors si s'obren dues instàncies del programa amb diferents presentacions. La darrera sobreescriu el PresentationManager de la primera i apareixen inconsistències molt greus.

En detectar l'error de disseny s'ha canviat la semàntica del PresentationManager, que ha passat a ser un controlador. Totes les crides remotes es fan des d'aquí. Així es pot arribar a implementar un sistema de memòria cau més interessant que simplement la vida dels objectes (tot i que això és fora de l'abast del projecte), encapsula l'estat de l'acció de presentar i evita redundàncies en abstreure les crides remotes. El desenvolupador només ha de cridar a PresentationManager::getAllParticipants() per recuperar-los, en comptes d'haver d'escriure la crida remota sencera. Permet, també, tenir operacions de granularitat més fina pels objectes locals que aprofitin el resultat de les crides remotes, de granularitat més gruixuda.

Diagrama de classes

Per diferenciar clarament la frontera del sistema s'ha indicat en gris els objectes del client i en blanc els del servidor.

A més de les classes explicades anteriorment, queden les implementacions d'ISimpleEventHandler. S'instancien des d'EyemeetingApplication per enviar notificacions als usuaris sobre determinats esdeveniments. En elles es defineix el missatge que s'envia, el tipus, si requereix confirmació, etc.



Il·lustració 57: Diagrama de classes

Restriccions

- Hi ha d'haver un i només un orador. El model permet que n'hi hagi zero perquè es necessita PresentationManager abans que l'usuari importi una presentació i, per tant, es determini el seu tipus.
- La data d'obertura d'una presentació és posterior a l'actual.

Aclariments

- Les operacions exposades a la capa de presentació només poden tenir un

paràmetre, que per conveni s'anomena *params*, de tipus matriu. La justificació es troba en el disseny de l'operació `eyeos::callMessage`, fora de l'abast d'aquest projecte.

- La classe `Participant` no té les especialitzacions de cada tipus de participant per mantenir un disseny el més simple possible. Calia aplicar el patró estat i inicialment també semblava convenient aplicar el patró factoria, però d'haver-ho fet així s'hauria complicat extremadament la lògica de l'aplicació sense aportar grans beneficis. Aquesta classe s'utilitzaria majoritàriament amb les llistes de la capa de presentació. El model d'aquests components no accepta objectes, només matrius i, per tant, fa que la classe `Participant` sigui molt poc utilitzada.

Diagrames de seqüència

S'ofereixen els diagrames de seqüència més importants.

`PresentationManager`

`NextSlide`, `previousSlide`, `getSlide`, `getBasePath`, `getNumSlides`, `getCurrentSlide`,
`getFilePath`, `getHash`, `setCurrentSlide`, `setPresentationProperties`,
`updateParticipants`, `updatePresentation`, `getCurrentUserPermissions`

Es deleguen en la classe corresponent.

`__populateParticipants`

Operació trivial que recull la resposta del servidor i crea les estructures necessàries en el client per poder utilitzar els participants.

`getAllParticipants`

Operació trivial que demana els participants al servidor i els enregistra amb `__populateParticipants`.

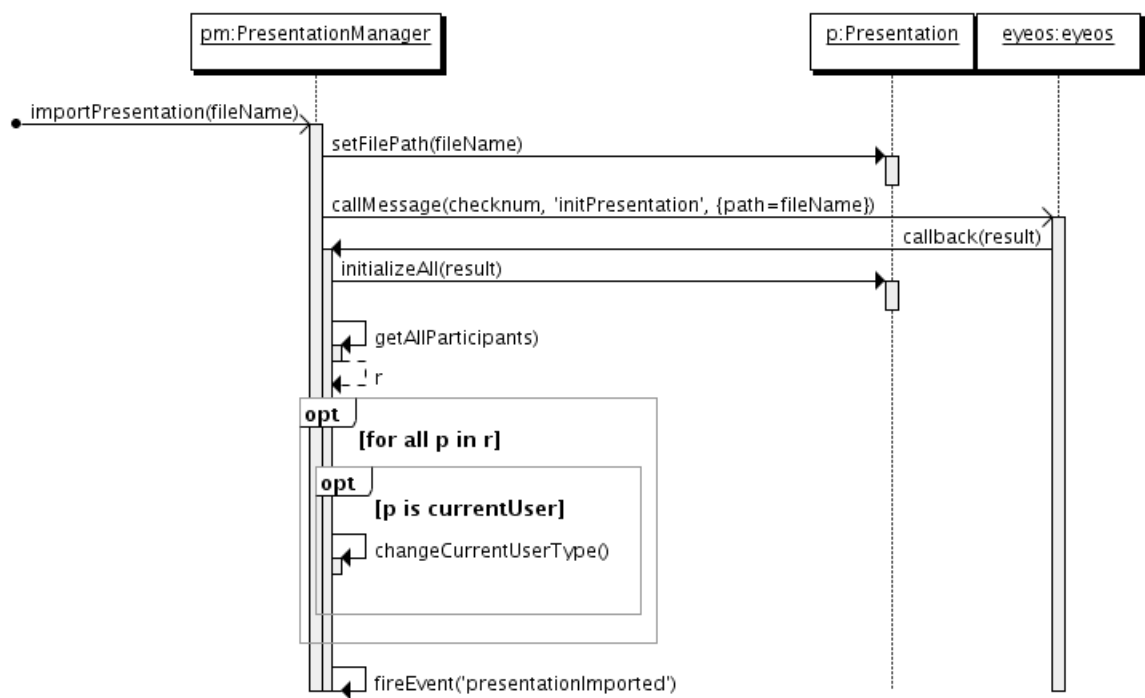
`getAllContactsAndWorkGroups`

Operació trivial que demana els participants i els seus grups de treball al servidor.

`getAllWorkgroupsByUser`

Operació trivial que demana els grups de treball de l'usuari al framework `Workgroups` del servidor.

importPresentation

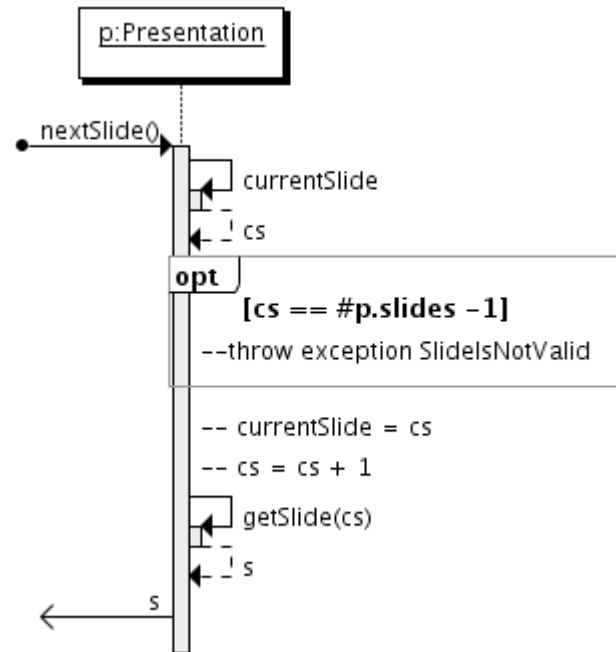


Il·lustració 58: Diagrama de seqüència d' importPresentation

Presentation

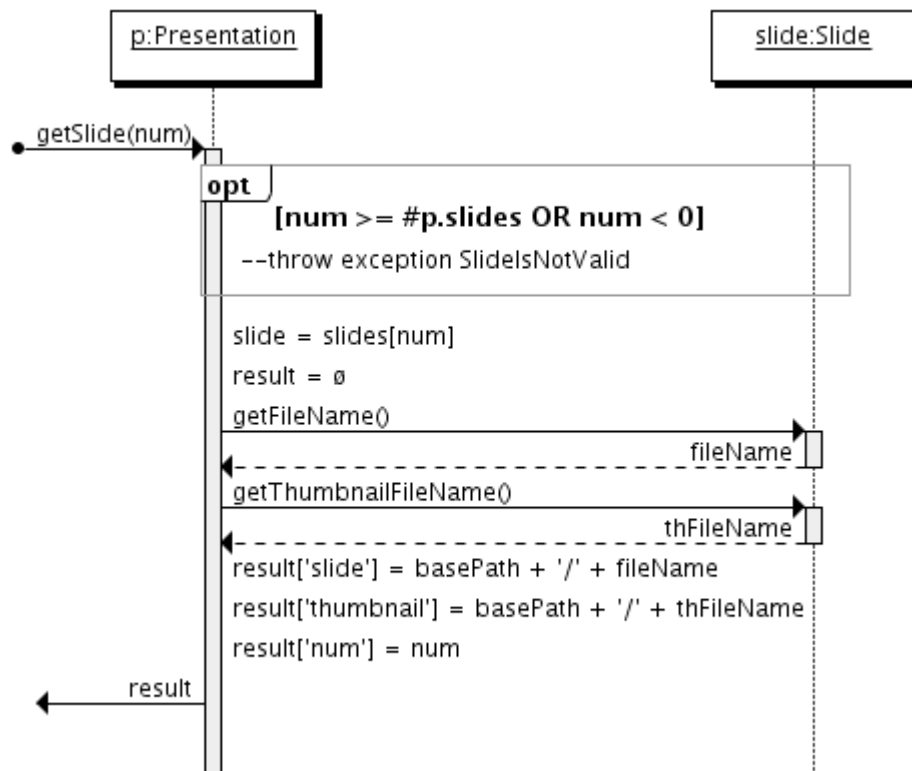
NextSlide i previousSlide

Il·lustració 59



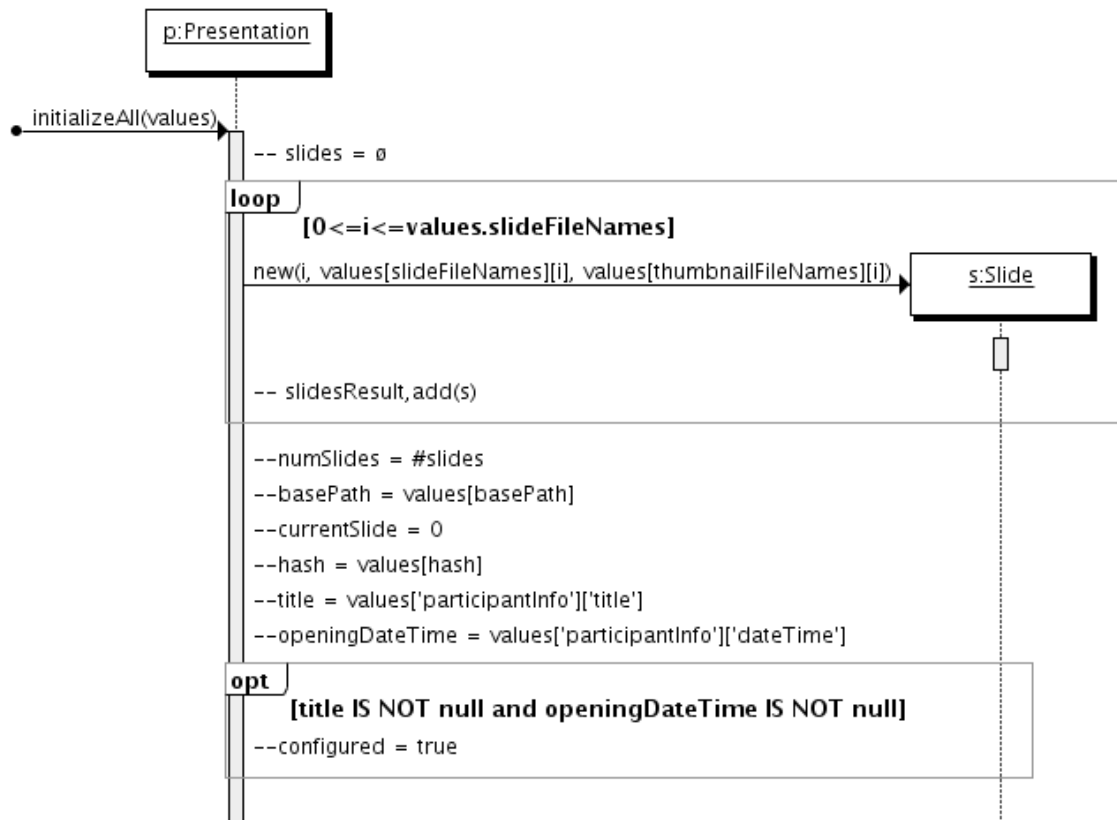
Il·lustració 59: Diagrama de seqüència de `nextSlide`

getSlide



Il·lustració 60: Diagrama de seqüència de `getSlide`

initializeAll



Il·lustració 61: Diagrama de seqüència d' `initializeAll`

Les classes `Participant` i `Slide` només tenen *getters* i *setters* sense cap interès.

En el servidor hi ha el controlador i els esdeveniments:

EyemeetingApplication

En el servidor no hi ha *callbacks* en general i el funcionament és molt més similar al conegut Java. La gran diferència és que en cada crida que arriba al controlador s'ha d'assumir que no hi ha cap classe viva en el node.

__run

Operació obligatòria per restriccions de disseny de les aplicacions en l'eyeOS. S'hi declaren tots els fitxers JavaScript que s'han de servir al client.

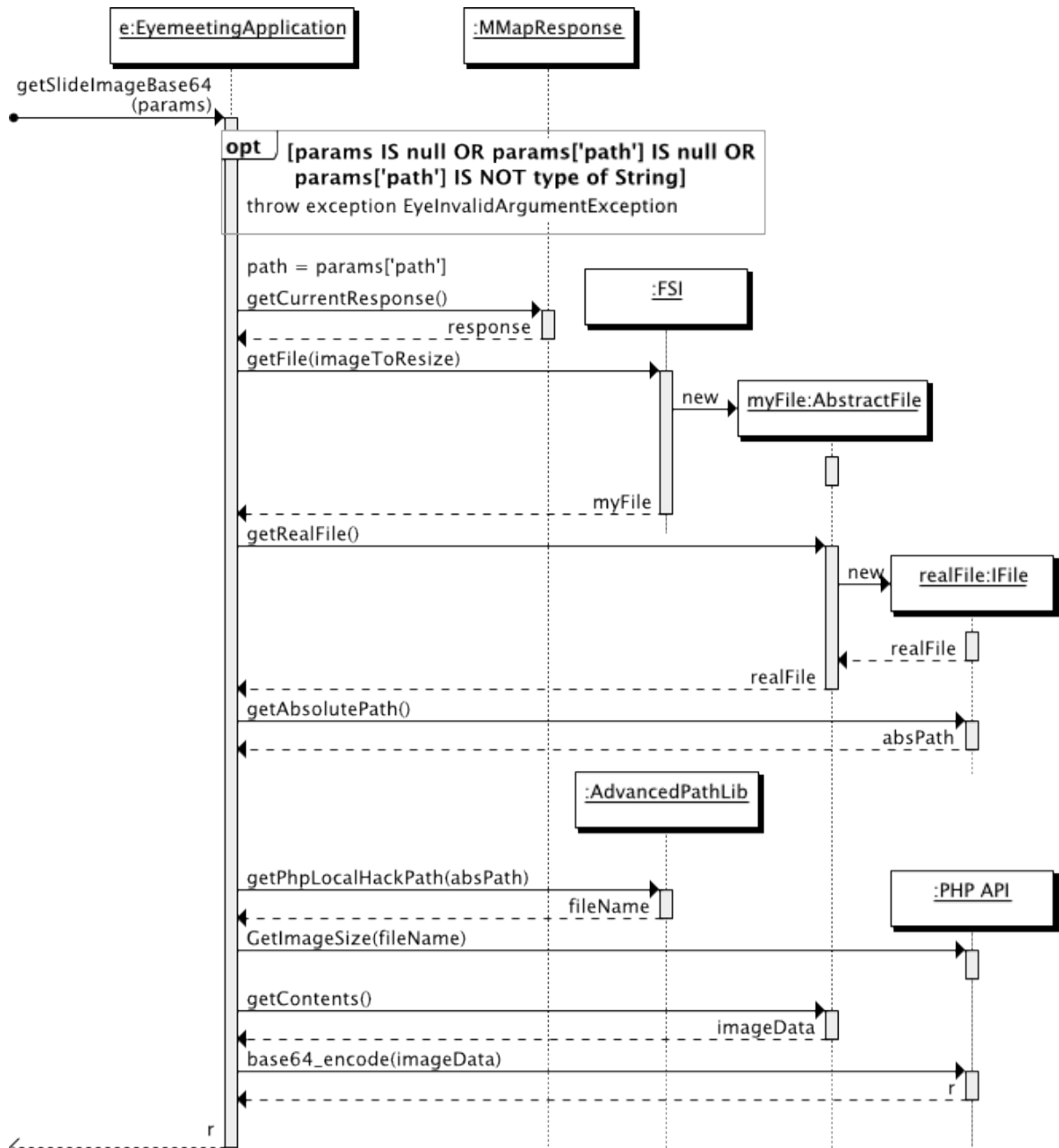
createThumbnails

Crea les miniatures de les diapositives.

El diagrama de seqüència és molt gran i no té gran interès. L'operació crea les

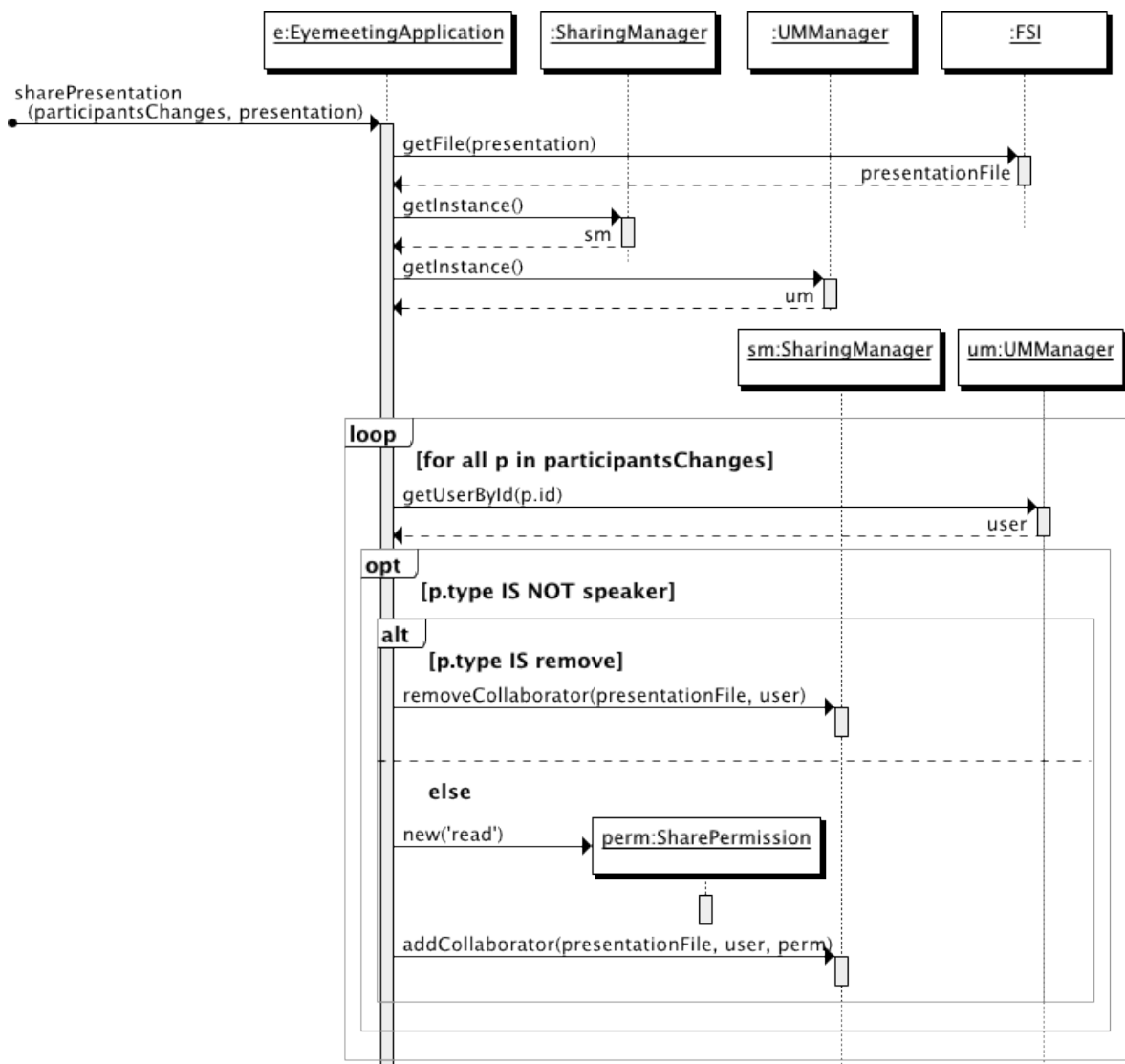
imatges petites i les anomena d'acord amb el vector de diapositives que ha entrat com paràmetre.

getSlideImageBase64



Il·lustració 62: Diagrama de seqüència de `getSlideImageBase64`

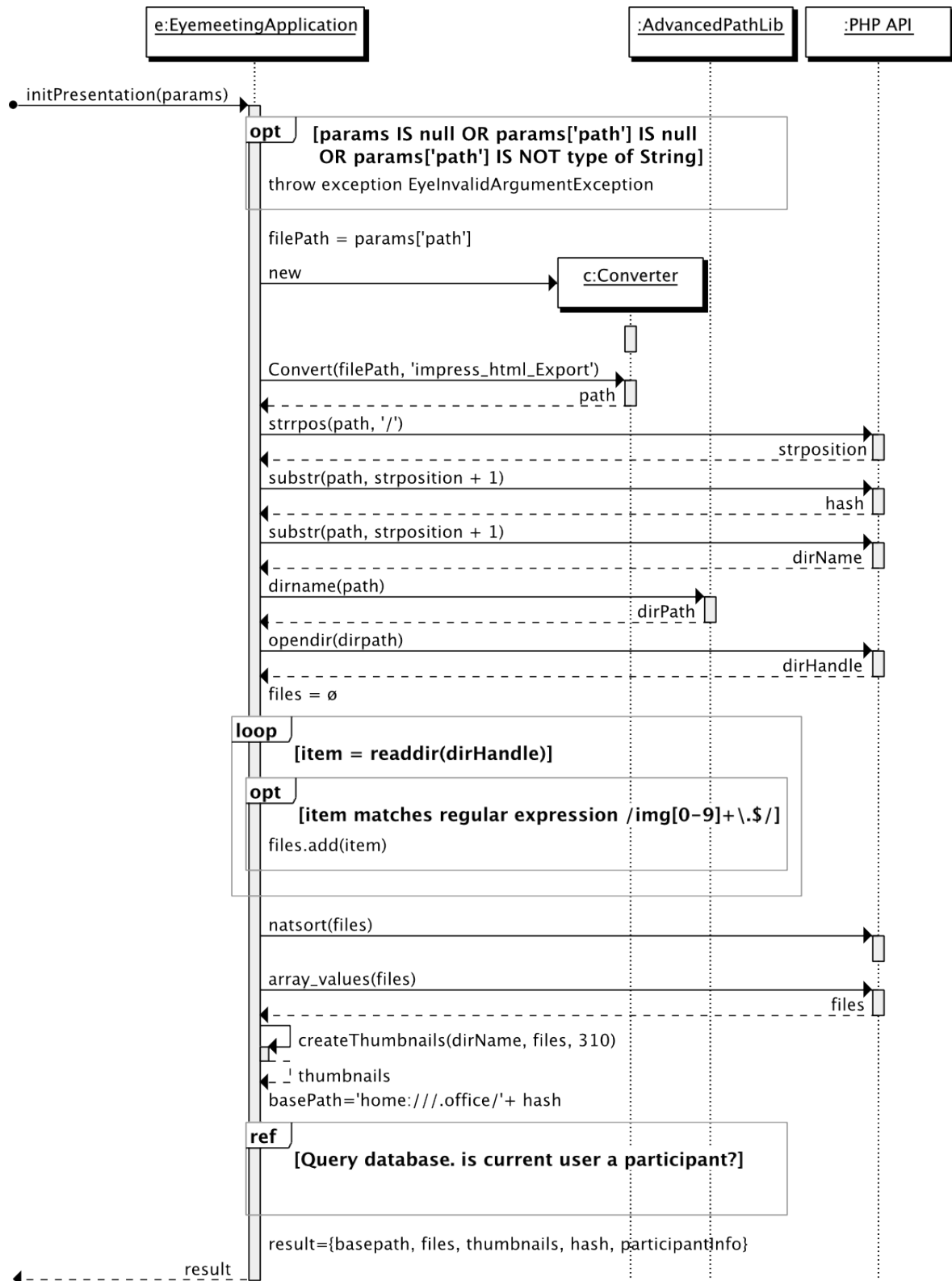
sharePresentation



Il·lustració 63: Diagrama de seqüència de `sharePresentation`

initPresentation

La seqüència d'iniciar una presentació és potser de les més senzilles però a la vegada importants en l'aplicació. La Il·lustració 64 mostra com s'instancia el convertidor de l'OpenOffice i com s'enregistra cada imatge generada per cada diapositiva en la posició correcta d'un vector. S'utilitza l'algorisme *natsort* disponible en l'API de PHP.



Il·lustració 64: Diagrama de seqüència d'initPresentation

getAllContactsAndWorkGroups

Operació trivial que utilitza els ExecModules de l'eyeOS per recuperar els contactes i els seus grups de treball.

updateParticipants

Operació similar a updatePresentation que actualitza les dades dels participants, n'afegeix de nous o n'esborra. Finalment comparteix la presentació i n'envia les invitacions.

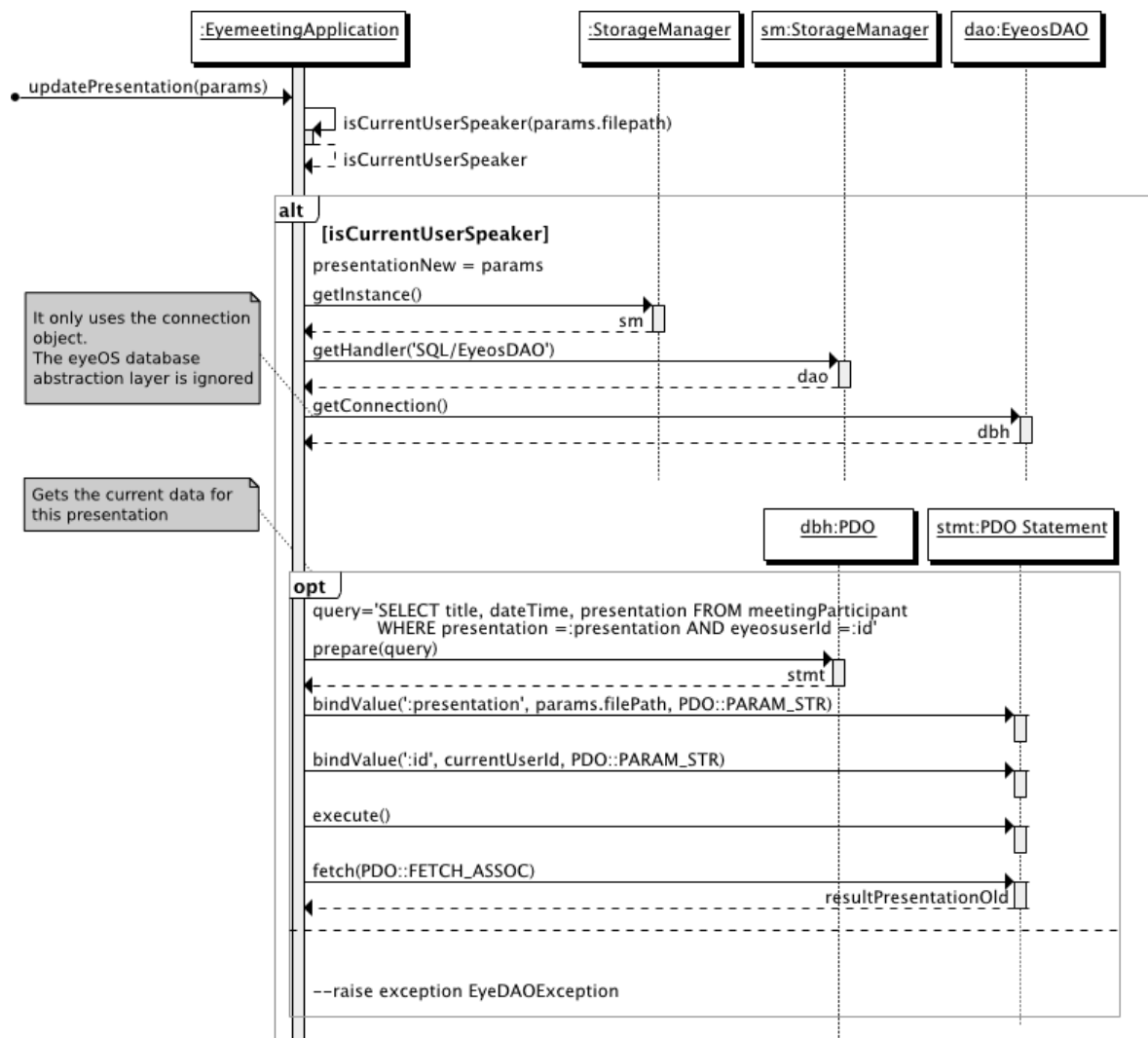
updatePresentation

Operació que utilitza la capa d'abstracció de dades del PHP per actualitzar la base de dades i enregistrar-hi les noves dades de la presentació.

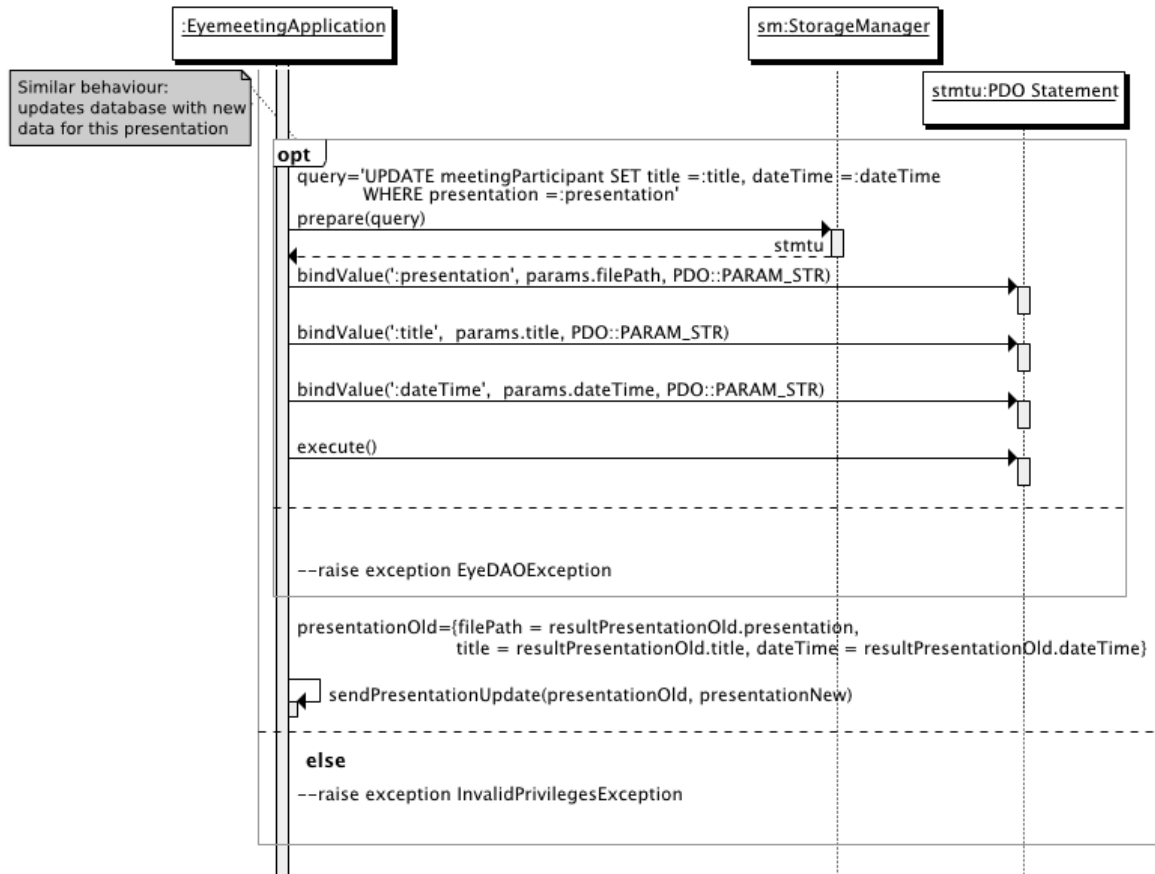
En aquest diagrama s'ha suposat que ja existeix una variable `currentUserId` per simplificar-lo. Obtenir l'identificador de l'usuari en eyeOS és una operació trivial però llarga de navegació a partir d'una instància de ProcManager i sense cap interès per il·lustrar el funcionament del mètode.

Per errors de disseny de l'eyeOS no s'ha pogut utilitzar la seva capa d'abstracció de dades, que és com habitualment es desenvolupen les seves aplicacions. Ha calgut ignorar-la i anar una mica més avall, fins les eines d'abstracció de dades de PHP: PDO¹⁹. En el capítol 5.2.2.4 Disseny de la capa d'accés a dades s'explica en detall.

¹⁹ PHP Data Objects és una extensió de PHP que defineix una interfície lleugera i consistent per accedir a bases de dades amb PHP.



Il·lustració 65: Diagrama de seqüència d' updatePresentation (1)



Il·lustració 66: Diagrama de seqüència d' *updatePresentation* (2)

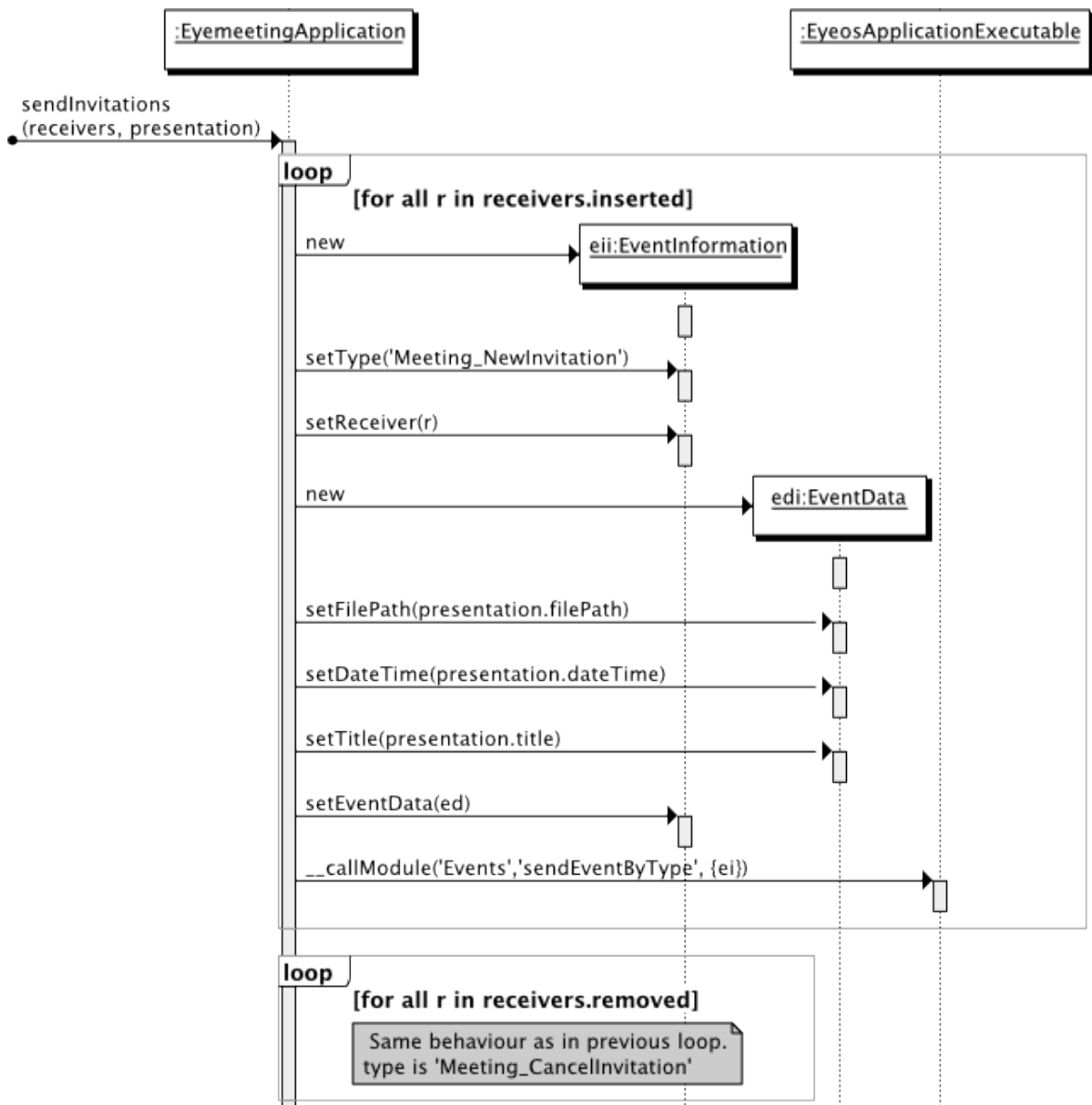
getParticipants

Operació trivial que recupera els participants de la base de dades. Es delega en la capa d'abstracció de dades que proporciona PHP.

sendPresentationUpdate

Operació similar a *sendInvitations*.

sendInvitations



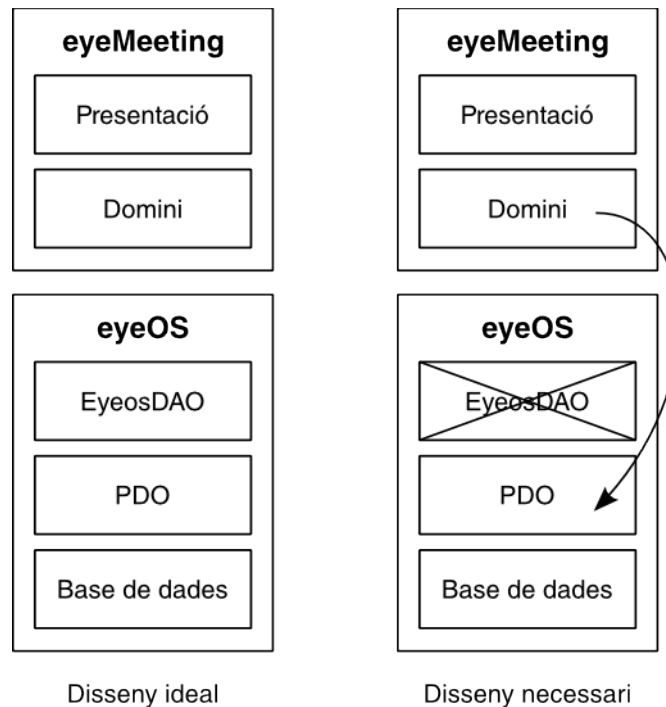
Il·lustració 67: Diagrama de seqüència de `sendInvitations`

ISimpleEventHandler

Les classes que implementen aquesta interfície simplement redefeixen el mètode `autoFill` per enviar les dades desitjades al sistema d'esdeveniments. La lògica és pràcticament idèntica per les 4 implementacions.

5.2.2.4 Disseny de la capa d'accés a dades

Habitualment es parla de disseny en 3 capes tot i que no és requisit que siguin 3. El nombre de capes pot ser major o menor segons trobi convenient el dissenyador.



Il·lustració 68: Comparació entre dissenys ideal i necessari de la capa d'abstracció de dades

L'eyeOS és un sistema en 3 capes però la majoria de les seves aplicacions en tenen només dues i utilitzen la capa d'abstracció de dades de l'eyeOS. Aquesta capa fa les interaccions amb la base de dades i retorna objectes que encapsulen els resultats. Dins l'eyeOS s'anomenen *nomdelataulaTransferObject*. En el moment de desenvolupar aquesta aplicació la capa de d'abstracció de dades de l'eyeOS no estava suficientment ben feta i no s'ha pogut emprar. No obstant, sí que s'ha pogut utilitzar PDO, una altra capa d'abstracció de dades de PHP, en un nivell inferior. Degut a l'error de disseny en l'eyeOS, és necessari instanciar l'EyeosDAO i obtenir-ne l'objecte de connexió a la base de dades, fet que provoca que l'aplicació encara depengui de la capa problemàtica.

Inicialment existia la classe *meetingparticipantTransferObject*, però era necessari que la base de dades tingués una clau primària composta i l'EyeosDAO no ho permet. En detectar l'error (extern) es va canviar el disseny per adaptar-lo a l'entorn, encara que va anar en contra de fer codi reusable o extensible, per exemple.

Disseny de la base de dades

L'esquema conceptual conté classes d'objectes. L'eyeOS en què s'ha

desenvolupat l'eyeMeeting funciona amb un SGBD relacional (MySQL), que implementa taules relacionals amb el motor InnoDB. Fer el disseny lògic d'aquesta base de dades és traduir els elements de l'esquema conceptual a components implementables pels SGBD relacionals. No es dissenya la base de dades des de zero, sinó només els components que necessita l'eyeMeeting i que aniran a la mateixa base de dades que l'eyeOS.

Hi ha dues entitats que han de ser enregistrades a la base de dades: Participant i Presentation. Aquestes han de ser persistents entre sessions de l'eyeOS. No cal que ho sigui la classe Slide, que es crea cada cop que s'importa una presentació.

L'aproximació típica seria gestionar les especialitzacions de Participant seguint algun patró (*class table inheritance*, *concrete table inheritance* o *single table inheritance*), crear una taula per Presentation i fer les claus primàries i foranes corresponents.

Tractament de la jerarquia en Participant

En la Taula 2 hi ha les estratègies possibles [ES2], [Fowler]:

Estratègia	Avantatges	Desavantatges
Class Table Inheritance	Simple Canviable	Poc eficient (múltiples accessos per objecte)
Concrete Table Inheritance	Eficient (un accés per objecte)	Poc canviable (propagació de canvis fets a les superclasses)
Single Table Inheritance	Eficient (un accés per objecte) Canviable	Pèrdua d'espai (però la BD pot ajudar)

Taula 2: Estratègies de tractament de la jerarquia

Les classes de la jerarquia de participant no tenen atributs i, per tant, no té sentit traduir-les com una taula. A més a més, habitualment es fan servir tots els participants.

Alguns atributs de la classe Presentation són derivats de la importació del fitxer odp o ppt(x). D'acord amb els diagrames de seqüència de l'inici de l'aplicació, no s'han d'enregistrar.

- Participant(id, nickname, type) id és clau forana a eyeosuser(id)
- Presentation(filePath, title, openingDateTime, numSlides,, currentSlide, basePath, hash, configured)

S'ha decidit, per tant, col·lapsar tota la jerarquia de Participant amb l'estratègia *single table inheritance*.

Desnormalització

Una observació prèvia al disseny pot fer incrementar notablement el rendiment: pràcticament mai s'utilitzen les dades de Presentation i de Participant de manera independent i es fan pocs canvis que afectin a més d'un registre. Només quan un orador canvia els detalls de la presentació s'actualitzarien diversos registres i no s'espera que això sigui freqüent. Un disseny senzill, normalitzat, de taules Participant (o les de la seva jerarquia) i Presentation obligaria a fer *joins* pràcticament sempre, amb el cost tan elevat que tenen.

És convenient desnormalitzar [DABD] un disseny quan:

- El *join* és una operació freqüent
- No s'esperen gaires canvis

S'ha decidit, doncs, desnormalitzar el disseny fins a la forma Boyce-Codd:

meetingparticipant(eyeosuserId.presentation, participantType, dateTime, title)

- eyeosuserId és clau forana a eyeosuser(id)
- presentation és la ruta del fitxer de la presentació

Vegeu Codi 1: SQL de l'eyeMeeting

```

10 SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
11
12 -- Estructura de la taula `meetingparticipant`
13
14 CREATE TABLE IF NOT EXISTS `meetingparticipant` (
15   `eyeosuserId` varchar(128) NOT NULL,
16   `presentation` varchar(128) NOT NULL,
17   `participantType` varchar(15) NOT NULL,
18   `dateTime` datetime NOT NULL,
19   `title` varchar(255) NOT NULL,
20   PRIMARY KEY (`eyeosuserId`,`presentation`)
21 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
22 [...]
23 --
24 -- Restriccions per la taula `meetingparticipant`
25 --
26 ALTER TABLE `meetingparticipant`
27   ADD CONSTRAINT `meetingparticipant_ibfk_1` FOREIGN KEY
28   (`eyeosuserId`) REFERENCES `eyeosuser` (`id`);

```

Codi 1: SQL de l'eyeMeeting

6 IMPLEMENTACIÓ

6.1 Entorn de desenvolupament²⁰

6.1.1 Instal·lació

Consulteu Bibliografia i referències: eyeOS 2.3, 2.4, 2.5 Installation Manual.

L'equip de l'eyeOS garanteix el seu funcionament en Debian i RedHat. El portàtil de treball utilitza Fedora 14.

No obstant, en la versió 2.3 va caldre reparar el nucli de l'eyeOS perquè fos compatible amb aquest entorn: la classe Converter (Converter.php) i ooo2any.py, els dos fitxers responsables de donar suport a documents d'oficina no funcionen amb un servidor LAMPP d'ApacheFriends i Fedora perquè les rutes són diferents. En concret, la línia 93 s'ha de canviar i afegir-hi la part en negreta:

```
93 shell_exec('LD_LIBRARY_PATH="/lib/::usr/lib"; LANG=en_US.utf-8;' .  
    $cmd);
```

Codi 2: Canvis en la crida al sistema de l'OpenOffice

6.1.2 Desenvolupament

Un entorn de desenvolupament típic consisteix en:

- *Instal·lació LAMP*
Linux-Apache-MySQL-PHP. Consulteu el manual d'instal·lació.
- *Mozilla Firefox 4 o superior, branca estable.*
Es poden utilitzar els perfils per simular diversos usuaris simultàniament. Cal executar el Firefox amb els paràmetres '-no-remote -P'
`firefox -no-remote -P`
Ctrl + shift + R per netejar la memòria cau. Cada cop que s'edita un fitxer JavaScript, cal refrescar l'eyeOS i netejar la memòria cau.
- *Firebug*
La pestanya de consola, script i l'inspector són eines freqüents.
- *Netbeans (+ PHP plug-in)*
Una manera habitual de treballar és crear un projecte PHP amb codi existent (tots els fitxers de l'eyeOS) i després emprar només la carpeta de l'aplicació que s'està desenvolupant. Així és més senzill examinar l'API.

²⁰ [Gamonal]

- *Subversion*

Per controlar les versions de l'aplicació, no per utilitzar la branca inestable de l'eyeOS.

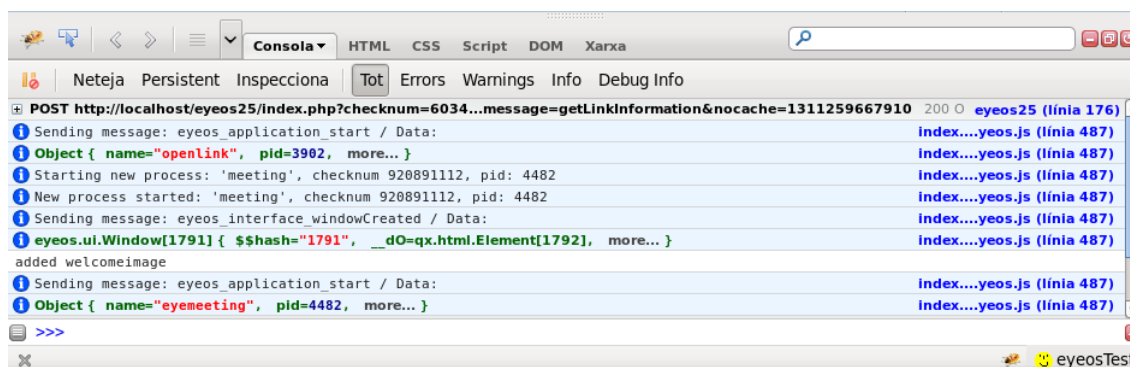
Una estratègia freqüent per desenvolupar una aplicació és crear una carpeta en /apps/ i baixar el codi de trunk/ de l'aplicació allà, de manera que tots els canvis en l'aplicació, i només en aquella aplicació, seran publicats a trunk/.

```
cd your/path/to/eyeOS/apps/
svn co https://yourRepo.com/trunk appName
```

6.1.3 Depuració

Consola i registre

És extremadament útil no només per depurar codi JavaScript, sinó per veure com s'executen les peticions o per llegir els errors i excepcions que es disparen des del servidor.

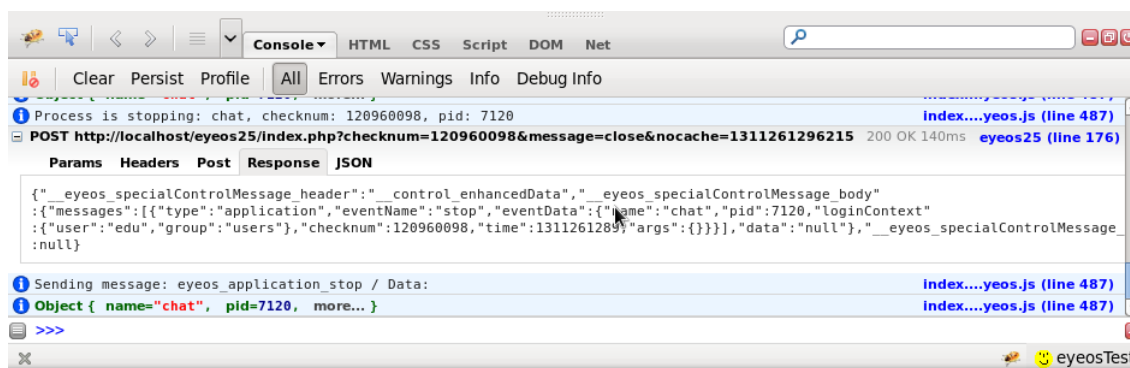


Il·lustració 69: Consola del Firebug

Per escriure a la consola del Firebug cal cridar **console.log("hello world");** des de qualsevol lloc en el codi JavaScript. S'hi poden passar tants arguments com siguin necessaris. El Firebug els fusionarà en una línia. Per exemple, **console.log(2,4,6,8,"foo",bar)**. També existeixen **console.debug**, **console.info**, **console.warn**, i **console.error**.²¹

Per veure quines dades s'estan transferint en qualsevol crida al servidor, cal fer clic en el símbol + al costat de la línia POST.

²¹<http://getfirebug.com/logging>

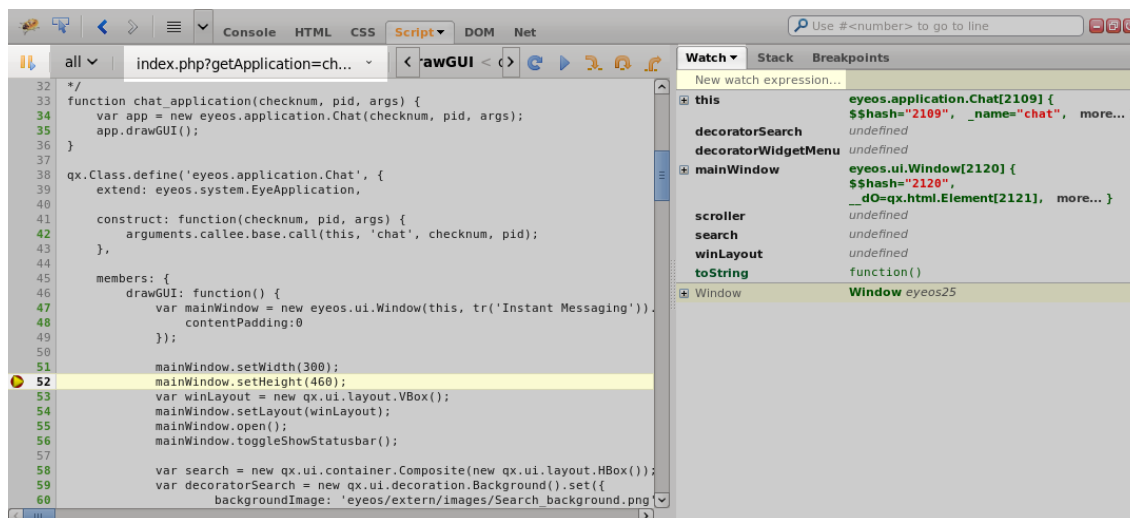


Il·lustració 70: Vista de les dades POST en la consola del Firebug

Pestanya script

En la pestanya *script* es pot veure el codi que s'executa. Les classes de Qooxdoo estan comprimides però el codi de les aplicacions es pot llegir còmodament. Es pot fer clic en qualsevol línia o escriure **debugger**; per establir-hi un punt d'interrupció.

Quan l'execució s'ha interromput es poden mirar expressions.



Il·lustració 71: Pestanya d'script i punt d'interrupció en el Firebug

És bona idea filtrar el codi per aplicació. Quan s'executa per primera vegada, el seu codi JavaScript s'afegeix al cos de la pàgina i queda disponible al Firebug.

Registres de l'eyeOS

Quan la configuració de l'eyeOS ho permet, tota l'activitat queda registrada en fitxers. Alguns errors comuns que es poden detectar amb els registres són:

- Consultes incorrectes a la base de dades.
- Permisos incorrectes amb el FSI²² (eyeOS).
- Permisos incorrectes del sistema (p. ex. GNU/Linux).
- No es pot trobar els binaris de l'OpenOffice.
- etc

Les versions estables de l'eyeOS no tenen activat el registre d'activitat per omissió.

Activar el registre

Es fa en dos passos:

1. Editar settings.php i canviar el valor «release» a «debug»

```
//RELEASE OR DEBUG
define('SYSTEM_TYPE', 'release');
```
2. Editar el fitxer de configuració dels registres en
eyeos/system/conf/libs/log4php/configuration.xml. Es pot sobreescriure pel
que hi ha disponible en trunk/²³

Llegir el registre

Els fitxers del registre són en /eyeos/system/conf/libs/log4php/logs . Es pot veure instantàniament el que s'hi escriu:

```
$ tail -f *.log
```

Escriure el registre

En el codi PHP cal instanciar un *Logger*

```
$logger = Logger::getLogger('apps.myapp');
$logger->info('this is a line');
```

També hi ha altres mètodes disponibles, com **\$logger->debug** i **\$logger->warn**

6.2 Exemples d'ús de les tecnologies utilitzades

S'han seleccionat algunes de les funcions més representatives del projecte i de la tecnologia que utilitza per mostrar-les amb detall.

²² FSI – File System Interface [Gamonal]

²³ <https://code.eyeos.org/trunk/>

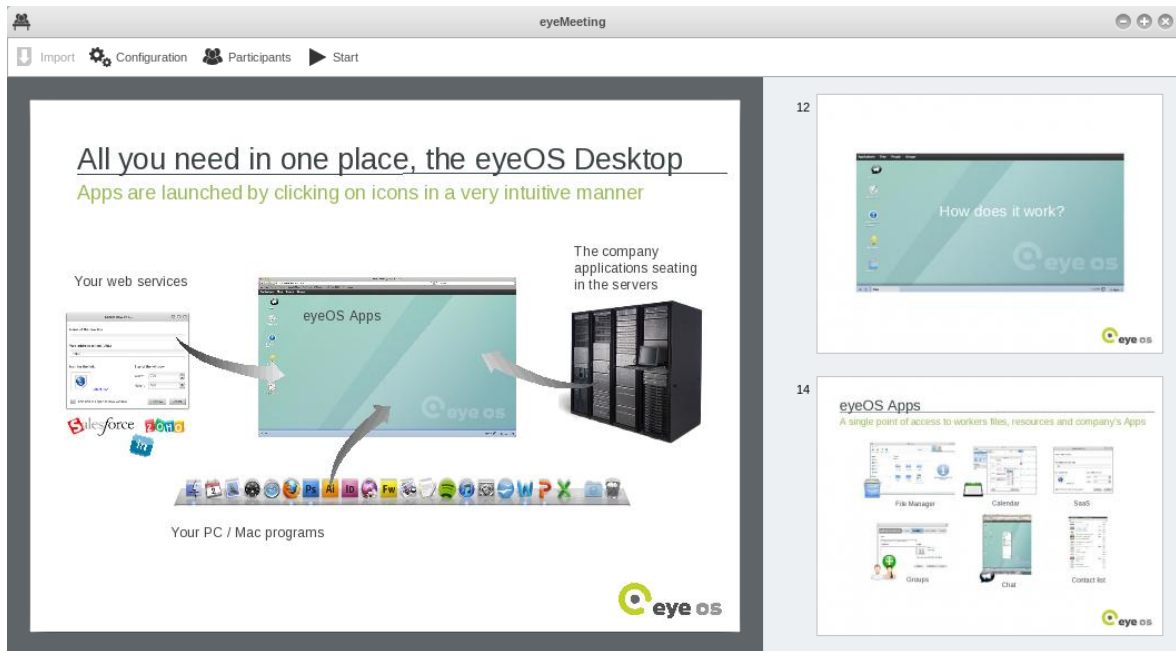
6.2.1 Creació de la finestra i dels seus components

El fragment següent mostra el codi d'eyemeeting.js que crida el primer mètode de la interfície gràfica (línies 1..4), la constructora i una part del mètode que crea tota la vista (línies 51..88). La interacció és senzilla: crea una finestra, la configura amb un Layout, crea un contenidor que allotjarà els components i l'afegeix a la finestra. Tots els contenidors han de tenir un Layout. Window és també un contenidor.

```
1 function eyemeeting_application(checknum, pid, args) {
2     var app = new eyeos.application.eyemeeting(checknum, pid);
3     app.drawGUI();
4 }
5
6 qx.Class.define("eyeos.application.eyemeeting", {
7     extend : eyeos.system.EyeApplication,
8
9     construct : function(checknum, pid) {
10         arguments.callee.base.call(this, "meeting", checknum, pid);
11         this.checknum = checknum;
12     },
13
14     drawGUI : function() {
15         this.window = new eyeos.ui.Window(this, "eyeMeeting",
16 'index.php?extern=/images/16x16/apps/meeting/icon.png').set({
17             contentPadding: 0
18         });
19         this.window.setLayout(new qx.ui.layout.VBox(0));
20         this.window.setAllowMaximize(true);
21         this.window.open();
22
23         /*
24          * main area
25          */
26         var mainContainer = new qx.ui.container.Composite();
27         mainContainer.setLayout(new qx.ui.layout.HBox());
28         mainContainer.set({minWidth: 50, minHeight: 50});
29         mainContainer.setPadding(0);
30         mainContainer.setBackgroundColor('#5e6468');
31
32         this.window.add(mainContainer, {flex:1});
33     }
34 });
```

Codi 3: Finestra principal

El codi sencer genera una vista com la de la Il·lustració 72.



Il·lustració 72: Finestra principal de l'eyeMeeting

6.2.2 Suport per documents d'oficina

El suport per documents d'oficina és una funció crítica en aquest projecte. L'eyeMeeting utilitza l'eyeOS per convertir un fitxer en format odp o ppt(x) a imatges. Aquesta funció està repartida en diverses operacions.

S'inicia quan l'usuari fa clic en el botó d'importar un fitxer, en eyemeeting.js:

```

192 /*
193  * callback of click on import button
194  *
195  * @param e Event
196  */
197 _onImport : function(e) {
198   this._fileChooser.showOpenDialog(this.window, function(choice, path) {
199     if (choice == eyeos.dialogs.FileChooser.APPROVE_OPTION) {
200       this._presentationManager.importPresentation(path);
201     }
202   }, this, 'home:///');
203 },

```

Codi 4: Callback en fer clic al botó d'importació

En PresentationManager es fa la crida remota que utilitza l'OpenOffice:

```

115 importPresentation : function(fileName) {
116     this.getPresentation().setFilePath(fileName);
117     eyeos.callMessage(this.getChecknum(), 'initPresentation',
        { path : fileName }, function(result) {
118         this.getPresentation().initializeAll(result);
        [...]
154     }, this, {timeout: 60000}); // openoffice might be slow
155 },

```

Codi 5: Crida remota per convertir un document d'oficina

Al servidor, EyemeetingApplication en eyemeeting.php utilitza la classe Converter, de l'eyeOS:

```

59 /**
60  * Converts de presentation from odp to images with OpenOffice,
61  * creates the thumbnails and loads related info, if any.
62  *
63  * @returns array of basePath, slideFileNames, thumbnailFileNames,
        hash, participantInfo
64  * @param params.path is the path of the file home://~user/file.odp
65  */
66 public function initPresentation($params) {
71     $filePath = $params['path'];
72     $myConverter = new Converter();
74     $path = $myConverter->Convert($filePath, 'impress_html_Export');
    [...]
115     return array('basePath' => $basePath, 'slideFileNames' => $files,
        'thumbnailFileNames' => $thumbnails, 'hash' => $hash,
        'participantInfo' => $rq);
116 }

```

Codi 6: Conversió de documents d'oficina -- costat del servidor

El mètode Convert retorna la ubicació dels fitxers fruit de la conversió, que és /users/usuari/files/.office/hash/. Per relacionar cada fitxer amb la diapositiva corresponent, s'aprofita els noms de fitxer consecutius que retorna OpenOffice i s'enregistren en una matriu que s'ordena amb *natsort*, de manera que queden com si ho hagués ordenat una persona seguint el sentit comú (p. ex. img1.jpg, img2.jpg, ..., img10.jpg).

El directori .office, per una presentació de 4 diapositives queda així:

```
[edu@fedora-laptop .office]$ ls
5076c938fd1cd3d67181a5ef2a571cae

[edu@fedora-laptop 5076c938fd1cd3d67181a5ef2a571cae]$ ls -lo
total 15
-rw-r--r-- 1 apache 464 11 ago 17:06
5076c938fd1cd3d67181a5ef2a571cae
-rw-r--r-- 1 apache 5239 11 ago 17:06 img0.jpg
-rw-r--r-- 1 apache 3115 11 ago 17:06 img0-th.jpg
-rw-r--r-- 1 apache 558 11 ago 17:06 img1.html
-rw-r--r-- 1 apache 5528 11 ago 17:06 img1.jpg
-rw-r--r-- 1 apache 3293 11 ago 17:06 img1-th.jpg
-rw-r--r-- 1 apache 535 11 ago 17:06 img2.html
-rw-r--r-- 1 apache 12917 11 ago 17:06 img2.jpg
-rw-r--r-- 1 apache 5562 11 ago 17:06 img2-th.jpg
-rw-r--r-- 1 apache 535 11 ago 17:06 img3.html
-rw-r--r-- 1 apache 17853 11 ago 17:06 img3.jpg
-rw-r--r-- 1 apache 8109 11 ago 17:06 img3-th.jpg
-rw-r--r-- 1 apache 549 11 ago 17:06 text0.html
-rw-r--r-- 1 apache 577 11 ago 17:06 text1.html
-rw-r--r-- 1 apache 605 11 ago 17:06 text2.html
-rw-r--r-- 1 apache 583 11 ago 17:06 text3.html
[edu@fedora-laptop 5076c938fd1cd3d67181a5ef2a571cae]
```

Codi 7: Sortida del terminal amb la llista de fitxers generats per l'OOo

img*.jpg i text*.html són generats per l'OpenOffice. img*-th.jpg les genera el codi de l'eyeMeeting.

L'eyeOS esborra automàticament els continguts d'aquesta carpeta en tancar la sessió.

6.2.3 SVG

El component que representa les diapositives en la interfície gràfica és un tros de codi SVG en mig de l'HTML generat. Cal que sigui així perquè es necessita una representació de traços fets a mà alçada que es pugui crear per composició amb un cost baix. Si s'hagués fet amb altres estratègies, per exemple fusionant imatges de la mida de la diapositiva, hauria estat molt més costós i poc extensible. Tancaria la porta a editar les pintades, entre altres coses. El codi es va creant dinàmicament a mesura que l'usuari sol·licita diapositives al sistema.

En un sol component SVG hi ha un grup per cada diapositiva. Cada grup conté 3 grups més: la diapositiva, les pintades dels altres participants i les pintades de l'usuari. Els grups en SVG poden funcionar com si fossin capes, tot permetent una separació semàntica. A més a més, incloure la imatge en un grup amb l'etiqueta

d'imatge de l'svg permet no haver-se de preocupar de les transformacions que es facin en tot el llenç (*zooming*, *paning*, etc). En el Codi 8 s'han omès alguns atributs per facilitar-ne la lectura.

```
1 <svg version="1.1" baseProfile="full" id="slideCanvas" >
2   <image xlink:href="index.php?extern=images/meeting/welcome.png"
      height="480" width="640" display="none"/>
3   <g id="meeting-slide-0">
4     <g id="slide">
5       <image xlink:href="data:image/jpg;base64,[...]"
      height="480" width="640"/>
6     </g>
7     <g id="participantsDoodles"/>
8     <g id="myDoodles">
9       <path fill="none" stroke="green" stroke-width="5" stroke-
      linecap="round" d="M292,55H291 [...] 368,73H369L371, 71"/>
10    </g>
11  </g>
12 </svg>
```

Codi 8: Estructura del node SVG

La línia 3 obre el primer grup per la diapositiva 0. Quan l'usuari demani veure la diapositiva següent, s'afegeix l'atribut "display:none" en el grup de la diapositiva 0 i es crea una estructura idèntica a les línies 3..10 però amb els continguts de la diapositiva 1.

Cal tenir en compte que les línies 5 i 9, que contenen una imatge serialitzada i les coordenades del camí que ha pintat l'usuari, estan retallades per poder-les encabir en aquest document. Com es pot imaginar, una línia complexa en SVG pot ser extremadament llarga. L'eyeOS té limitacions en la quantitat de dades que es poden enviar amb NetSync a altres usuaris i això fa que a partir de certa complexitat no es pugui compartir la pintada. La reparació d'aquest problema queda fora de l'abast del projecte.

Per generar aquesta estructura s'utilitza el component `svg.Embed.Svg` des d'`eyemeeting.js` (Codi 9).

```

125 this.slideView = new svg.embed.Svg();
126 this._slideViewSvg = this.slideView.getSvg();
127 this.slideView.setHeight(479);
128 this.slideView.setWidth(639);
129 var welcomeImage = new svg.struct.Image();
130 welcomeImage.setHref('index.php?extern=images/meeting/welcome.png');
131 welcomeImage.set({
132   height: 480, width: 640
133 });
134 this._slideViewSvg.add(welcomeImage);
135 this._slideViewSvg.addListener("mousedown", this._startDraw, this);
136 this._slideViewSvg.addListener("mousemove", this._draw, this);
137 this._slideViewSvg.addListener("mouseup", this._endDraw, this);
138
139 // properties of the svg element are kept in sync with the svg widget
140 // so width and height are overriden by the flex 1 property the
    container has
141 this._slideViewSvg.set({ id: 'slideCanvas'});

```

Codi 9: Creació del component `svg.embed.Svg`

En `drawGUI` es crea el component i a mesura que es van rebent les diapositives, es completa. En la línia 474 es decideix el comportament en funció de si la diapositiva ja s'havia demanat abans o no (Codi 10 i 11)

```

459 * browse those tags, attempt to find the one with the next slides.
460 * if found, show it. create it otherwise.
461 *
462 */
463 var svgDom = this._slideViewSvg.getDomElement();
464 var svgDomChildren = qx.dom.Hierarchy.getChildElements(svgDom);
465 var targetGroup = null;
466 // set all slides to not visible except the new one (current)
467 for (var i = 0, l = svgDomChildren.length; i < l; ++i) {
468   svgDomChildren[i].setAttribute('display', 'none');
469   if (svgDomChildren[i].id == 'meeting-slide-' +
        this._presentationManager.getCurrentSlide()) {
470     targetGroup = svgDomChildren[i];
471   }
472 }
473
474 if (targetGroup) {
475   targetGroup.setAttribute('display', 'inline');
476
477   fadeIn.start();
478   this.CHANGING_SLIDE = false;
479 } else {

```

Codi 10: Creació dels components gràfics de les diapositives (1)


```

480 // couldn't find a group, new slide has to be requested
482
483 eyeos.callMessage(this.getChecknum(), 'getSlideImageBase64',
    {path : newSource['slide']}, function(result) {
484     fadeIn.start();
485
486     var slideWrapper= new svg.struct.Group(); // the slide.
487
488     var groupSlide = new svg.struct.Group(); // wraps slide's picture
489     var groupParticipantsDoodles = new svg.struct.Group(); // wraps
other's doodles
490     var groupMyDoodles = new svg.struct.Group(); // wraps my doodles
491
492     slideWrapper.setId('meeting-slide-' +
        this._presentationManager.getCurrentSlide());
493     groupSlide.setId('slide');
494     groupParticipantsDoodles.setId('participantsDoodles');
495     groupMyDoodles.setId('myDoodles');
496
497     var slideImg = new svg.struct.Image();
498     slideImg.setHeight(480);
499     slideImg.setWidth(640);
500     slideImg.setHref('data:image/jpg;base64,' + result);
501
502     groupSlide.add(slideImg);
503     slideWrapper.add(groupSlide);
504     slideWrapper.add(groupParticipantsDoodles);
505     slideWrapper.add(groupMyDoodles);
506     this._slideViewSvg.add(slideWrapper);
507
508     this.slideView.setHeight(480);
509     this.slideView.setWidth(640);
510
511     this.slidesStack[this._presentationManager.getCurrentSlide()] =
        {slide: groupSlide, participantsDoodles:
groupParticipantsDoodles, myDoodles: groupMyDoodles };
[... ]
520     this.CHANGING_SLIDE = false;
521 }, this);
522 }
[... ]
540 }

```

Codi 11: Creació dels components gràfics de les diapositives (2)

Atès que mostrar diapositives és una funció molt important del programa, s'ha volgut mostrar més en detall.

En la segona part del codi (línia 483) se sol·licita la diapositiva nova al servidor. En la callback es creen els components necessaris per tenir el grup `<g id="meeting-slide-{n}">`.

slidesStack és una estructura auxiliar per poder seguir operant amb els objectes un cop generat el codi svg. CHANGING_SLIDE és un semàfor per evitar que l'usuari canviï les diapositives mentre hi ha un altre canvi en curs.

6.2.4 Crida remota

Una crida al servidor és tan simple com utilitzar l'objecte eyeos.

```
257 eyeos.callMessage(this.getChecknum(), 'updatePresentation', attrs,  
    function(result) {  
258     console.log('updated presentation. result: ' + result);  
259 });
```

Codi 12: Crida remota

La seva signatura és

```
callMessage: function (checknum, message, params, callback,  
context, options);
```

En general:

- *message* és el nom de l'operació remota.
- *params* és una matriu amb els paràmetres de l'operació remota
- *callback* és el nom de l'operació local que s'executarà en acabar l'operació remota. El seu paràmetre s'emplena amb el resultat de l'operació remota.
- *context* és l'objecte a què fa referència la paraula clau *this* en el callback
- *options* és un mapa que defineix com gestionar errors i missatges de control, com per exemple el timeout.

Una operació en el servidor ha de ser en el controlador (EyemeetingApplication) i declarada pública i estàtica:

```
public static function updatePresentation($params);
```

6.2.5 Listeners

Qooxdoo és una biblioteca gràfica molt similar a Swing o GTK. Està orientada a esdeveniments i per tant és habitual trobar-se *listeners* pel codi. Per exemple, per capturar les accions sobre un botó:

```
this._onChooseParticipants és el callback, que podria haver estat una funció  
this.participantsChooserButton.addListener("execute",  
    this._onChooseParticipants, this);
```

Codi 13: Listener

anònima, com el callback que es crida quan s'acaba el fadeOut en cada canvi de

diapositives:

```
415 fadeOut.addListener('finish', function() {  
[...]  
538 }, this);
```

Codi 14: Funció de callback anònima

S'utilitzen per tot: per rebre missatges del NetSync, del D-Bus, per formularis, per executar codi quan ha aparegut un component a la pantalla, en seleccionar... La documentació de Qooxdoo indica, per cada component, quins esdeveniments pot rebre. En el cas de les classes pròpies de l'eyeMeeting es poden disparar esdeveniments:

- classe.fireEvent('nom');
- classe.fireDataEvent('nom', paràmetres)

6.2.6 NetSync i d-bus

L'eyeOS imita un sistema operatiu i algunes parts del disseny segueixen els mateixos conceptes. Per exemple, hi ha identificadors de processos (però no processos pròpiament), permisos... i també un d-bus.

Les aplicacions es comuniquen amb el D-Bus, els usuaris es comuniquen amb NetSync.

El NetSync és un objecte en JavaScript que té també parts en el servidor. L'Apache utilitza ActiveMQ per organitzar les cues de missatges. Aquest aspecte és, però, transparent al programador, que només ha d'escriure això:

```
546 var netSync = eyeos.netSync.NetSync.getInstance();  
[...]  
549 var message = new eyeos.netSync.Message({  
550   type: 'eyemeeting',  
551   name: this._presentationManager.getHash(),  
552   to: 'eyemeeting_' + this._presentationManager.getHash(),  
553   data: {  
554     luggage: 'a sample string'  
555   }  
556 });  
557 netSync.send(message);
```

Codi 15: Enviament de missatge amb NetSync

Un missatge de NetSync pot enviar-se a un canal o a un usuari individual. En el codi d'exemple s'envia al canal

eyemeeting_5076c938fd1cd3d67181a5ef2a571cae

Cada missatge té un tipus i un nom, que formen l'etiqueta identificativa.

En importar una presentació de l'eyeMeeting, l'usuari se subscriu al seu canal:

```
222 // netsync: subscribe and listen
223 var netSync = eyeos.netSync.NetSync.getInstance();
224 var syncTag = 'eyemeeting_' + this._presentationManager.getHash();
225 netSync.subscribe(syncTag);
226 var bus = eyeos.messageBus.getInstance();
227 this.idBusListener = bus.addListener('eyeos_' + syncTag,
    this._onNetSyncMessageReceived , this);
```

Codi 16: Alta al canal de NetSync

i en tancar l'aplicació, se'n dóna de baixa i esborra el *listener* del D-Bus

```
681 // unsubscribe from channel
682 var syncTag = 'eyemeeting_' + this._presentationManager.getHash();
683 netSync.unsubscribe(syncTag);
684
685 // remove the listener from dbus. it's a system-wide singleton
686 eyeos.messageBus.getInstance().removeListenerById(
    this.idBusListener);
```

Codi 17: Baixa del canal de NetSync

El D-Bus és un *singleton* per tot l'eyeOS i, per tant, en tornar a obrir el programa no es restableix res. Se segueix utilitzant la mateixa instància que el primer cop que s'ha obert el programa. Els listeners anteriors hi romandran si no s'esborren amb l'operació `removeListenerById`.

6.2.7 Compartició de pintades

Per compartir pintades cal enviar missatges de NetSync al canal amb les dades de la pintada serialitzades. Això es fa tot just en acabar-la:

```
729 _endDraw : function() {
730     this.sendDoodle( qx.xml.Element.serialize(
        this._path.getDomElement()));
731     this._pathdata = null;
732 },
```

Codi 18: Acabament de la pintada

En `sendDoodle` es crea el missatge amb les dades adequades i s'envia:

```

564 sendDoodle : function(doodle) {
565     if ((this._presentationManager.isCurrentUserSpeaker() ||
        this._presentationManager.isCurrentUserActiveAttendee()) &&
        this._presentationManager.isRunning()) {
566         var netSync = eyeos.netSync.NetSync.getInstance();
567         var slide = this._presentationManager.getCurrentSlide();
568
569         var message = new eyeos.netSync.Message({
570             type: 'eyemeeting',
571             name: this._presentationManager.getHash(),
572             to: 'eyemeeting_' + this._presentationManager.getHash(),
573             data: {
574                 from: eyeos.getCurrentUserData().id ,
575                 action: 'doodle',
576                 doodle: {doodle: doodle, slide: slide}
577             }
578         });
579         netSync.send(message);
580     }
581 },

```

Codi 19: Enviament de la pintada

Quan es rep el missatge, isSpeaker o isActiveAttendee indiquen si el remitent és d'algun d'aquests tipus.

```

367 if (isSpeaker || isActiveAttendee) {
368     if (this._presentationManager.isRunning() && action == 'doodle') {
369         this.addDoodle(message['doodle']['doodle'],
            message['doodle']['slide']);
370     }
371 }

```

Codi 20: En rebre un missatge

i addDoodle és un mètode senzill que deserialitza la pintada, troba el node adequat i la hi afegeix:

```

745 addDoodle : function(doodle, slide) {
746     var node = qx.xml.Document.fromString(doodle);
747     var targetGroup = this.slidesStack[slide]['participantsDoodles'];
748     qx.dom.Element.insertEnd(node.documentElement,
        targetGroup.getDomElement());
749 }

```

Codi 21: Afegir una pintada

6.2.8 Servei de fitxers JavaScript al client

La implementació en el servidor té un mètode obligatori en el controlador de tota aplicació en l'eyeOS:

```

46 public static function __run(AppExecutionContext $context,
    MMapResponse $response) {
47     $buffer = '';
48     $basePath = EYE_ROOT . '/' . APPS_DIR . '/eyemeeting/';
49     $buffer .= file_get_contents($basePath .
        'model/js/PresentationManager.js');
50     $buffer .= file_get_contents($basePath .
        'model/js/Presentation.js');
    [...]
56     $response->appendToBody($buffer);
57 }

```

Codi 22: Servei de fitxers JavaScript al client

__run és el primer mètode que es crida quan s'arrenca l'aplicació. La variable \$buffer acumula tot el codi que cal desplegar en el client.

6.2.9 Connexió amb la base de dades

Com s'indica en el capítol 5.2.2.4, la capa de dades de l'eyeOS no es pot fer servir quan la taula per consultar té una clau primària composta, com és el cas de l'eyeMeeting. S'utilitza PDO, que és molt similar al JDBC, per exemple. Si s'hagués utilitzat l'EyeosDAO el codi de sota tindria un objecte meetingparticipantTransferObject amb tots els camps de la taula i s'utilitzarien els mètodes read, readAll i similars de \$dao.

```

247 $dao = StorageManager::getInstance()->getHandler('SQL/EyeosDAO');
248 $dbh = $dao->getConnection();
    [...]
258 $presentation = $params['presentation'];
259 $participants = array();
260 try {
261     $query = "SELECT *
                FROM meetingparticipant
                WHERE presentation =:presentation";
262     $stmt = $dbh->prepare($query);
263     $stmt->bindValue(':presentation', $presentation['filePath'],
                PDO::PARAM_STR);
264     $stmt->execute();
265     $participants = $stmt->fetchAll(PDO::FETCH_ASSOC);
266 } catch (Exception $e) {
267     throw new EyeDAOException("Couldn't get participants for this
                presentation", 0, $e);
268 }

```

Codi 23: Connexió amb la base de dades

7 PROVES DEL SISTEMA

7.1 Disseny dels jocs de proves i resultats

Les proves de l'aplicació han de mostrar que el programa compleix els requeriments. No es demostra formalment que el codi funcioni per raons pràctiques però sí que es demostra que l'aplicació fa el que ha de fer. Es mostren les proves tant pels requeriments funcionals com pels no funcionals.

Per cada prova s'indiquen els passos per executar-la i la condició de satisfacció. Si no s'indica el contrari, el resultat ha estat finalment positiu.

7.1.1 Proves funcionals

Iniciar presentació

1. L'orador fa clic al botó d'iniciar amb una presentació en curs.

Condicions de satisfacció:

- 'Orador ha pogut iniciar la presentació: controla quina diapositiva veuen els participants.
- La vista dels participants ha canviat i ja no tenen disponibles les miniatures per navegar lliurement per les diapositives.
- El sistema ha enviat un senyal START.
- Els participants se sincronitzen amb l'orador.

Acabar presentació

1. L'orador fa clic al botó d'acabar amb una presentació en curs.

Condicions de satisfacció:

- L'Orador ha pogut acabar la presentació: deixa de controlar quina diapositiva veuen els participants.
- La vista dels participants ha canviat i tenen disponibles les miniatures per navegar lliurement per les diapositives.
- El sistema ha enviat un senyal STOP.

Importar una presentació

1. L'usuari fa clic al botó d'importació.

2. L'usuari escull un fitxer en format odp o ppt(x) i prem OK.

Condicions de satisfacció:

- El sistema ha convertit un fitxer odp o ppt(x) en imatges i n'ha generat les miniatures.
- La vista ha canviat en funció de si la presentació estava iniciada o no i en funció del tipus d'usuari per la presentació triada.

Actualitzar la presentació

1. L'orador fa clic al botó de configuració.
2. L'orador entra el nou títol, data i hora i prem el botó d'OK.

Condicions de satisfacció:

- L'orador ha pogut canviar el títol, la data i l'hora de la presentació en curs.
- Els participants han rebut la notificació de l'esdeveniment.

Mostrar diapositiva

1. L'usuari fa clic en una miniatura.

i també:

1. L'orador canvia la diapositiva.
2. El sistema canvia la diapositiva dels participants.

Condicions de satisfacció:

- El participant ha pogut veure la diapositiva sol·licitada.
- Les miniatures s'han actualitzat.

Crear participant

1. L'orador fa clic al botó de participants.
2. L'orador escull un filtre en la primera llista.
3. L'orador fa doble clic en un element de la segona llista, que conté contactes filtrats.
4. L'orador escull el tipus de participant en el combobox de l'element corresponent en la tercera llista.

Condicions de satisfacció:

- L'orador ha pogut afegir un participant a la presentació.
- El participant s'ha afegit a la llista de participants i s'ha amagat de la llista

de contactes.

- El participant ha rebut la notificació de l'esdeveniment.
- Els permisos del fitxer de la presentació han estat actualitzats: l'usuari convidat pot llegir el fitxer.

Obtenir participant

1. L'orador fa clic al botó de participants.

Condicions de satisfacció:

- L'orador ha pogut veure els participants de la presentació.

Actualitzar participant

1. L'orador fa clic al botó de participants.
2. L'orador escull el tipus de participant en el combobox de l'element corresponent en la tercera llista.

Condicions de satisfacció:

- L'orador ha pogut canviar el tipus de participant de la presentació

Esborrar participant

1. L'orador fa clic al botó de participants.
2. L'orador fa doble clic al participant que vol esborrar.

Condicions de satisfacció:

- L'orador ha pogut esborrar un participant de la presentació.
- El participant s'ha esborrat de la llista de participants.
- El participant ha rebut la notificació de l'esdeveniment.

Crear pintada

1. El participant arrossega el ratolí sobre la diapositiva.
2. El participant deixa anar el ratolí (*mouse up*)

Condicions de satisfacció:

- El participant ha pogut pintar sobre la diapositiva que està veient.
- Si és participant actiu, ha pogut compartir la pintada i els altres participants l'han vista.

Obtenir pintada

1. Un altre participant actiu crea una pintada.

Condicions de satisfacció:

- El participant ha pogut rebre les pintades que fan altres participants actius.
- S'han mostrat les pintades d'altres participants en la diapositiva sobre la qual s'ha pintat originalment.

Tancar l'aplicació

1. L'usuari fa clic a la creu de la cantonada superior dreta.

Condicions de satisfacció:

- S'ha esborrat la subscripció al canal de NetSync de la presentació.
- S'han esborrat els *listeners* del D-Bus que ha creat l'eyeMeeting.
- Si l'usuari és l'orador, s'ha acabat la presentació.
- El programa s'ha tancat.

7.1.2 Proves de rendiment

Una aplicació distribuïda pot tenir problemes de rendiment per causes internes (solucionables raonablement) o externes, per les quals cal fallar gràcilment. S'han estudiat els dos problemes més grans en la xarxa: l'ample de banda i la latència.

L'ample de banda és la velocitat amb què arriben els continguts. La latència és el que es tarda en contestar. Així, és possible que una connexió de 50Mbps falli en enviar una imatge de només 10kb perquè tarda més en contestar que el temps definit com timeout en `eyeos::callMessage` (latència), o al revés, que l'usuari estigui utilitzant internet al mòbil en una zona sense 3G, amb HDSPA i GPRS, i per tant no pugui arribar una diapositiva o el codi necessari. Els talls de connexió es tracten com una latència infinita.

Ample de banda

1. L'usuari sol·licita una diapositiva nova al servidor.

Condicions de satisfacció:

- La diapositiva es mostra en un temps raonable proporcional a l'ample de banda disponible. Fins a 2 segons és acceptable.

Verificació

- El programa iprelay per GNU/Linux o configuracions particulars del servidor permeten simular poc ample de banda. S'ha utilitzat
`iprelay -b 128000 8002:localhost:80`
 Que simula una connexió de 128kbps per les peticions al port 8002. El programa supera les proves fins 64Kbps. Amb menys ample de banda no es pot servir l'eyeOS i per tant l'eyeMeeting no tindria sentit.

Latència de la xarxa

1. L'usuari sol·licita una diapositiva nova al servidor.

Condicions de satisfacció:

- La diapositiva es mostra en un temps raonable proporcional a l'ample de banda disponible. Fins a 2 segons és acceptable.

Verificació

- Les peticions al servidor amb `eyeos::callMessage()` tenen *timeout* per omissió que assegura una fallada adequada. S'assumeix que és suficientment gran per permetre un ús correcte del programa.

Temps d'importació d'una presentació

1. L'usuari importa una presentació.

Condicions de satisfacció:

- El sistema ha convertit la presentació en un temps raonable proporcional a la mida i el format del fitxer. Fins a 60 segons és acceptable.

Verificació

- Cal mesurar el temps que transcorre entre que es prem OK en la finestra d'importació del fitxer fins que apareix la primera diapositiva a la pantalla.
- Per més precisió es poden fer simulacions directes al servidor cridant a l'OpenOffice manualment:
 1. S'arrenca el servidor:


```
nohup soffice "-  
accept=socket,host=localhost,port=2002;urp;" >/dev/null  
2>/dev/null &
```
 2. i es crida:


```
ooo2any.py --extension eyeos --format impress_html_Export  
myfile.odp
```

Inicialment no s'ha superat aquest joc de proves. No hi ha recursos per fer més

ràpida la conversió i un fitxer de 30 diapositives amb una densitat i diversitat de color alta fa que apareguin timeouts. S'ha solucionat posant un timeout de 60 segons a la crida remota que sol·licita la conversió.

8 INSTAL·LACIÓ I EXECUCIÓ DEL SISTEMA

8.1 Requeriments

- Qooxdoo 1.3
- SVG Contrib 0.1
- Suport d'OpenOffice (hauria de generar imatges de 640*480px)
- eyeOS 2.5

8.2 Procediment

1. Descomprimiu el paquet de l'aplicació
2. Copieu-la en /apps/
3. Creeu /extras/meeting i copieu-hi apps/meeting/extras/meeting.sql
Executeu-lo a la base de dades.
4. Copieu /apps/meeting/extern/images/meeting a /extern/images i les icones corresponents.
5. Copieu /apps/meeting/Events/Handlers/Meeting a
/system/Frameworks/Events/Handlers
6. Compareu /apps/meeting/Events/Handlers/Events.php amb
/system/Frameworks/Events/Handlers/Events.php i afegiu en aquest darrer fitxer el que hi falti. No el sobreescriviu en cap cas. Altres aplicacions poden haver-lo modificat en crear més esdeveniments.
7. Podeu esborrar apps/eyemmeting/extern apps/eyemmeting/extras i
apps/eyemmeting/Frameworks

9 MANUAL DE L'USUARI

En aquest capítol hi ha el manual d'usuari que es distribueix amb l'aplicació, separat d'aquesta memòria.

9.1 Quant el manual

Copyright

Copyright © 2011 Eduard Gamonal Capdevila

Avís legal

Teniu permís per a copiar, distribuir i/o modificar aquest document, sota els termes de la Llicència de documentació lliure GNU (GFDL), versió 1.1 o qualsevol versió publicada posteriorment per la Free Software Foundation, sense seccions invariants, sense texts de portada i sense texts de contraportada. Podeu trobar una còpia de la GFDL en aquest [enllaç](#)²⁴.

EL DOCUMENT I LES VERSIONS MODIFICADES DEL DOCUMENT S'OFEREIXEN SOTA ELS TERMES DE LA LLICÈNCIA DE DOCUMENTACIÓ LLIURE DE GNU, TENINT EN COMPTE QUE:

1. EL DOCUMENT S'OFEREIX «TAL COM ÉS», SENSE CAP TIPUS DE GARANTIA, NI EXPLÍCITA NI IMPLÍCITA; AIXÒ INCLOU, SENSE LIMITAR-S'HI, LES GARANTIES QUE EL DOCUMENT O LA VERSIÓ MODIFICADA DEL DOCUMENT NO TINGUI DEFECTES, SIGUI COMERCIALITZABLE, SIGUI ADEQUAT PER A UN ÚS CONCRET O NO INFRINGEIXI CAP LLEI. TOT EL RISC PEL QUE FA A LA QUALITAT, EXACTITUD I RENDIMENT DEL DOCUMENT O LA VERSIÓ MODIFICADA DEL DOCUMENT ÉS VOSTRE. EN CAS QUE EL DOCUMENT RESULTÉS DEFECTUÓS EN QUALSEVOL ASPECTE, VÓS (NO PAS L'ESCRITOR INICIAL, L'AUTOR O CAP ALTRE COL·LABORADOR) ASSUMIU TOT EL COST DE MANTENIMENT, REPARACIÓ O CORRECCIÓ. AQUESTA RENÚNCIA DE GARANTIA CONSTITUEIX UNA PART ESSENCIAL D'AQUESTA LLICÈNCIA. NO S'AUTORITZA L'ÚS DE CAP DOCUMENT O VERSIÓ MODIFICADA DEL

²⁴ <http://www.gnu.org/copyleft/fdl.html>

DOCUMENT EXCEPTE SOTA AQUESTA RENÚNCIA DE GARANTIA; I

2. EN CAP CAS I SOTA CAP INTERPRETACIÓ LEGAL, JA SIGUI PER AGREUJAMENT (INCLOENT-HI LA NEGLIGÈNCIA), CONTRACTE O ALTRE CAS, L'AUTOR, L'ESCRITOR ORIGINAL, QUALSEVOL DELS COL·LABORADORS O DISTRIBUÏDORS DEL DOCUMENT O UNA VERSIÓ MODIFICADA DEL DOCUMENT NI CAP PROVEÏDOR D'AQUESTES PARTS NO SERAN RESPONSABLES DAVANT DE NINGÚ PER CAP DANY DIRECTE, INDIRECTE, ESPECIAL, ACCIDENTAL O CONSECUTIU DE QUALSEVOL TIPUS; AIXÒ INCLOU, SENSE LIMITAR-S'HI, ELS DANYS PER PÈRDUA DE CLIENTS, INTERRUPCIONS DE LA FEINA, FALLADA O MALFUNCIONAMENT DE L'ORDINADOR, O QUALSEVOL ALTRA PÈRDUA O DANY RELACIONAT AMB L'ÚS DEL DOCUMENT I LES VERSIONS MODIFICADES DEL DOCUMENT, FINS I TOT SI S'HA INFORMAT AQUESTA PART DE LA POSSIBILITAT D'AQUESTS DANYS.

Comentaris

Per a informar d'un error o fer algun suggeriment quant a l'eyeMeeting o aquest manual, contacteu amb els autors.

Historial de les revisions

Setembre 2011: Manual de l'eyeMeeting V1.0 (Eduard Gamonal Capdevila)

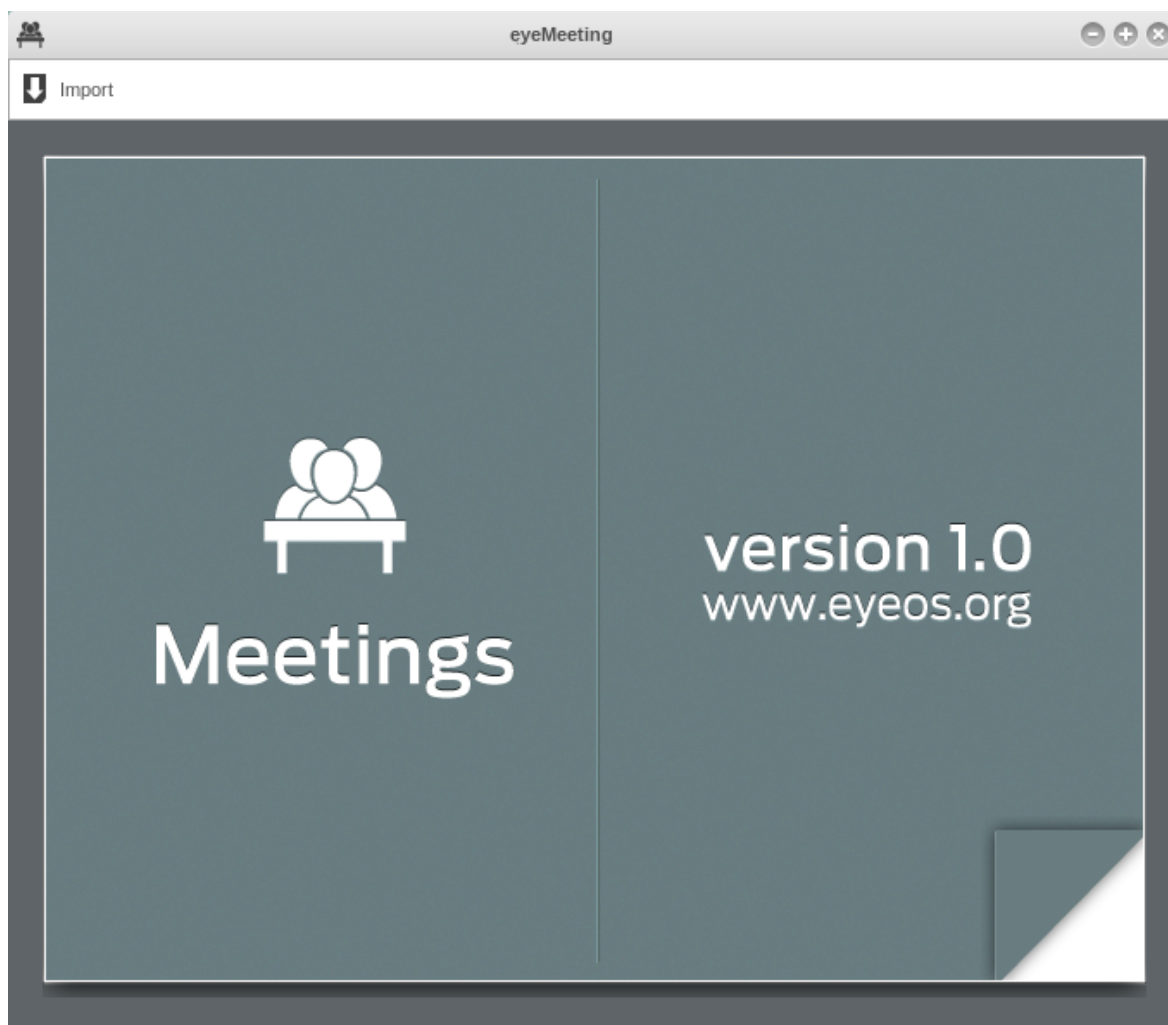
9.2 Introducció

L'eyeMeeting és una aplicació per l'eyeOS 2.5 o superior que us permet fer una reunió en línia amb una presentació. Un orador organitza les invitacions i executa la presentació. Els assistents poden, si l'orador ho permet, pintar sobre les diapositives. El canal de comunicació principal és recomanat és la veu, per telèfon o veu sobre IP, però l'aplicació no ho implementa.

9.2.1 Importar una presentació

En iniciar l'aplicació

En iniciar l'eyeMeeting es mostra la finestra següent:

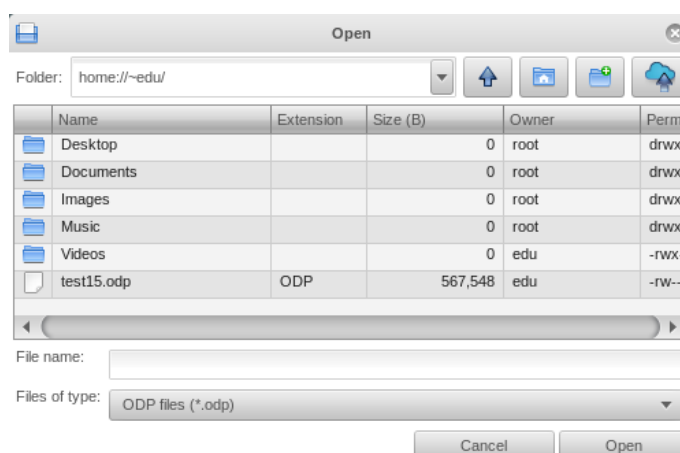


Il·lustració 73: Finestra inicial

Podeu importar una presentació.

En importar una presentació

En fer clic a **Import** es mostra la finestra següent:



Il·lustració 74: Navegador de fitxers

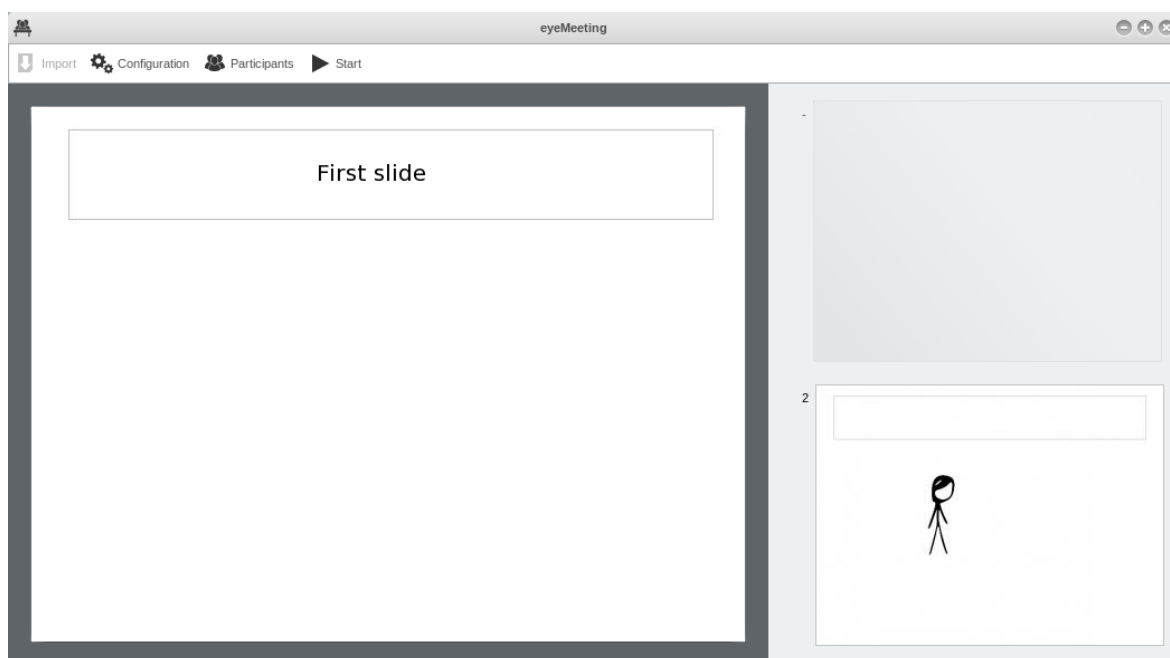
Trieu el fitxer de presentació que vulgueu importar.

Si voleu importar un fitxer vostre, cerqueu-lo en `home:///`. Si heu rebut una invitació d'un altre usuari, trobareu el fitxer que us indica la invitació en la carpeta compartida de l'usuari `share://usuari/`

Quan importeu un fitxer des del vostre `home:///` podeu esdevenir un orador i organitzar una reunió. Si l'importeu des de la carpeta compartida d'un contacte, sereu el tipus d'usuari que el vostre contacte hagi triat.

Quan hàgiu acabat, premeu **Ok**.

Si sou orador es mostrarà una pantalla com la de la Il·lustració 75

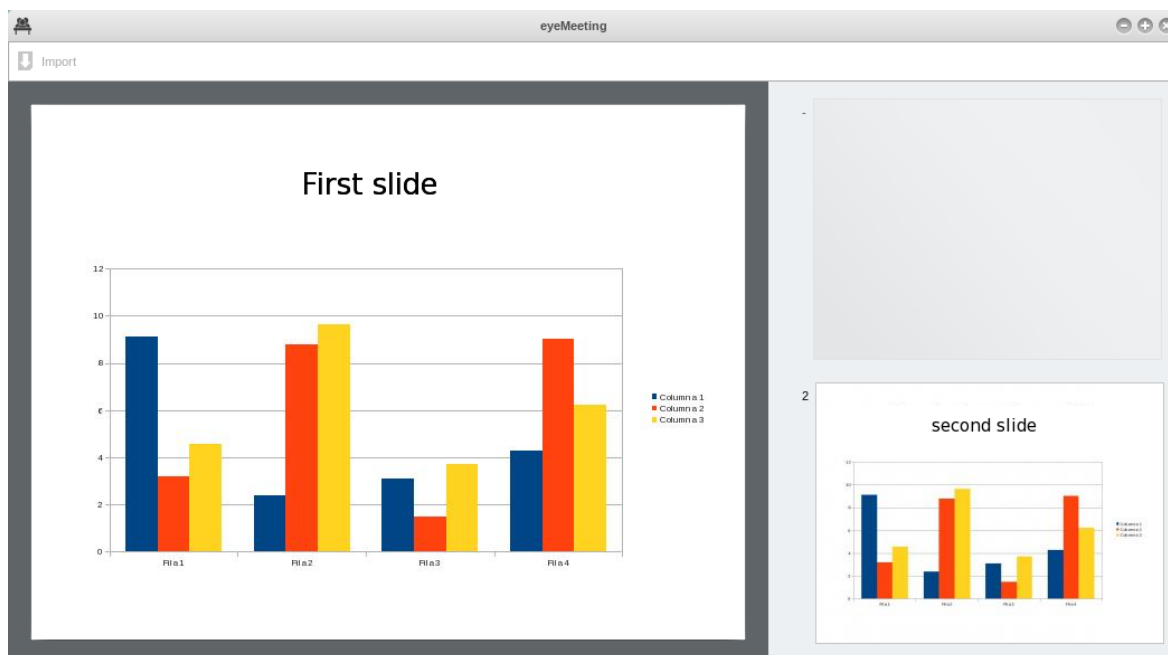


Il·lustració 75: Finestra de l'orador

La barra d'eines conté els elements següents:

- *Configuration*. Obre la finestra de configuració de la presentació.
- *Participants*. Obre la finestra de gestió dels participants.
- *Start*. Inicia la presentació. Només es mostra si la presentació no està iniciada.
- *Stop*. Acaba la presentació. Només es mostra si la presentació està iniciada.

Si sou un oient i la presentació no s'ha iniciat, es mostrarà la pantalla de la Il·lustració 76.



Il·lustració 76: Finestra de l'oient

Utilitzeu la imatge en miniatura de la següent diapositiva (inferior) per veure la següent diapositiva o la superior per veure l'anterior.

Configurar la presentació

Si sou orador podeu assignar un títol, data i hora recomanats per la presentació. Podeu començar la presentació en qualsevol moment però els participants assumiran que es farà en la data i hora que indiqueu.

The screenshot shows the eyeMeeting configuration window. It contains the following fields and controls:

- Title:** A text input field containing the text "I'm getting started with eyeMeeting".
- Date:** A date input field showing "Sep 16, 2011" with a calendar icon.
- Time:** A time input field showing "10:30".
- Submit:** A button to submit the configuration.

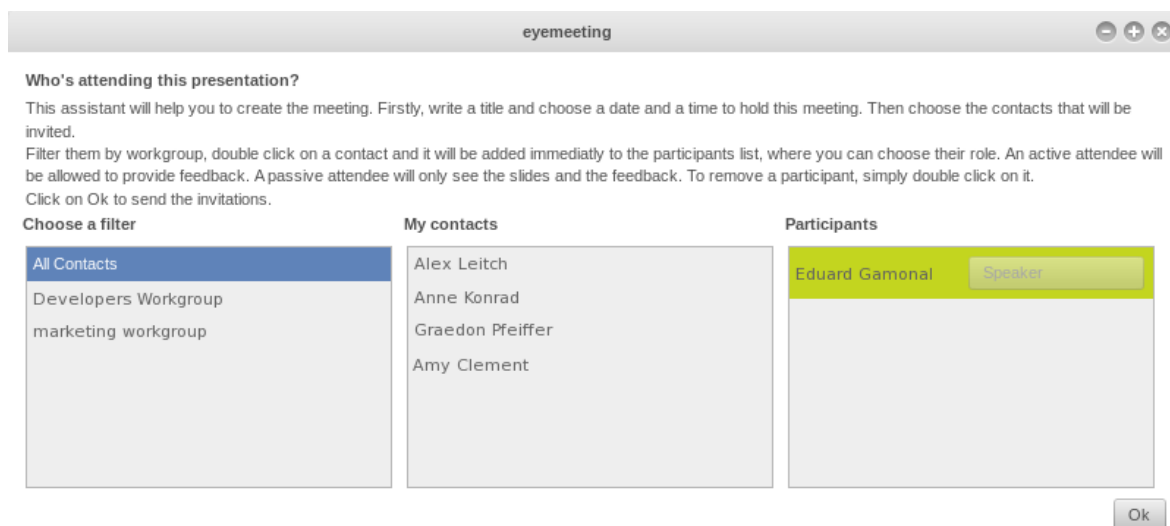
Il·lustració 77: Configuració

Entreu el títol, data i hora desitjats. La data i hora no poden ser anteriors al moment de prémer **Submit**. El sistema us alertarà si cometeu un error.

Quan hàgiu acabat, premeu **Submit**.

9.2.2 Gestionar els participants

Si sou orador podeu gestionar els participants de la presentació amb la finestra següent:



Il·lustració 78: Gestió dels participants

La finestra de gestió de participants conté els elements següents:

- **Filtre.** La primera llista, a l'esquerra, us permet seleccionar el filtre que s'aplicarà en la segona llista, de contactes. Els seus elements són «All» i els grups de treball de l'orador.
- **Contactes.** Us mostra els contactes filtrats amb el criteri de la llista de filtre. Feu doble clic en el nom d'un contacte per afegir-lo a la llista de participants.
- **Participants.** Us mostra els participants de la presentació. Feu doble clic en el nom d'un participant per esborrar-lo. Trieu el seu tipus en el menú desplegable al costat del seu nom.

Quan hàgiu acabat, premeu **Ok** per trametre els canvis i enviar les notificacions. Per tancar la finestra sense desar els canvis, premeu el botó amb una creu a la cantonada superior dreta.

9.2.3 Fer pintades

Podeu fer pintades a mà alçada sobre la diapositiva que es mostra. Si sou orador o oient actiu la pintada serà compartida automàticament i tots els participants la veuran.

Per fer una pintada arrossegueu el ratolí sobre el llenç de la diapositiva. En deixar anar el ratolí, s'intentarà compartir la pintada.

9.3 Autor

L'Eduard Gamonal Capdevila ha desenvolupat l'eyeMeeting, que funciona en l'eyeOS 2.5, i el manual d'usuari de l'eyeMeeting.

L'Equip de l'eyeOS ha desenvolupat la versió 2.5 de l'eyeOS.

9.4 Llicència

Aquest programa és programari lliure; podeu redistribuir-lo i/o modificar-lo sota els termes de la Llicència pública general de GNU tal com publica la Free Software Foundation; tant en la versió 2 de la llicència, o (a la vostra opció) una versió posterior.

Aquest programa es distribueix amb l'esperança de que sigui útil, però **SENSE CAP GARANTIA**; fins i tot sense la garantia implícita de **COMERCIALITZACIÓ** o **ADEQUACIÓ A UN ÚS CONCRET**. Per a més detalls vegeu la Llicència pública general de GNU.

S'inclou una còpia de la Llicència pública general de GNU en l'apèndix del manual de l'eyeMeeting. També podeu obtenir una còpia de la Llicència pública general de GNU de la Free Software Foundation visitant el seu lloc web o escrivint a

Free Software Foundation, Inc. 59 Temple Place - Suite 330 Boston , MA 02111-1307 USA

10 PLANIFICACIÓ REAL I VALORACIÓ ECONÒMICA

Com era d'esperar, hi ha hagut desviacions de certa importància en la previsió de les hores. Això ha estat degut a la inexperiència en iniciatives d'aquesta mida (mitigada lleugerament pel gestor de projectes d'eyeOS), no tenir en compte casos d'ús menors necessaris per l'arquitectura de l'eyeOS i una definició de l'abast inicial poc precisa. Això va portar a no concedir un nombre d'hores suficient a algunes tasques i a sobreestimar-ne d'altres, en especial les darreres.

- Fase d'inici

Definició del producte	15h
Formació	50h

- Fase de planificació

Planificació	15h
--------------	-----

- Fase d'execució

Especificació, disseny, implementació i proves del sistema base	150h
Especificació, disseny, implementació i proves del sistema de gestió d'usuaris	125h
Especificació, disseny, implementació i proves del sistema de compartició de pintades sobre les diapositives.	60h

- Fase de tancament

Memòria i preparació defensa	100h
------------------------------	------

Total 515h

En la fase d'inici es va discutir l'abast amb l'equip de disseny i el gestor de projectes. La proposta inicial incloïa funcions que més tard van considerar-se no oportunes pel producte final, com per exemple enquestes o un sistema per enviar comentaris als altres participants de la presentació. El producte va definir-se d'una manera més precisa i es va orientar a reunions amb presentació en comptes de presentacions de gran format.

En la fase d'execució van aparèixer problemes que no estaven previstos i que van provocar retards notables.

En la primera iteració:

- Dificultats amb l'eyeOS. Aquestes activitats no formen part del projecte però era crític resoldre-les per poder avançar.
 - Instal·lació complexa inesperada. Inclou suport per l'OpenOffice i va caldre corregir errors al nucli per fer-lo funcionar. Retard de 5 dies.
 - Error de disseny en el sistema de comunicació entre usuaris: NetSync té un error de disseny que a dia d'avui no es pot corregir i no estava documentat. L'error va aparèixer aquí i no en la segona iteració perquè part del sistema base és l'enviament de senyals per canviar diapositives. Retard de 3 dies.
- Disseny del sistema diferent de l'habitual. L'eyeOS és una RIA i les seves aplicacions són distribuïdes. Amb la formació que es tenia en aquest estadi el disseny va ser més propi d'una aplicació d'escriptori tradicional. Va caldre repensar-ho i hi va haver un retard de 3 dies.

Segona iteració

- Comportament no documentat d'algunes accions amb Qooxdoo. Les correccions formaven part, en aquell estadi, d'una versió més nova de Qooxdoo que l'eyeOS no podia fer servir. Va caldre implementar-ho d'una manera diferent i més segura. Retard de fins a 7 dies.
- Els components de Qooxdoo que calia utilitzar no es fan servir mai a l'oficina d'eyeOS i no hi havia coneixement al respecte. Retards petits i constants per errades menors en la implementació.
- Canvis en l'abast
 - Es demana afegir un sistema d'invitacions. La seva implementació requereix 4 dies.
 - Es decideix eliminar el sistema d'enviament de comentaris dels participants. S'avança 15 dies.

Tercera iteració

- L'ús del *contrib* SVG²⁵ i les converses amb el seu desenvolupador acceleren aquesta iteració i s'aconsegueix completar 8 dies abans.

En la fase de tancament no hi ha canvis.

10.1 Anàlisi de costos

Després de saber quant treball implica cada iteració es pot fer una valoració econòmica del projecte. Cal tenir en compte tant els recursos humans com els materials.

10.1.1 Recursos humans

El cost més alt del projecte és en els recursos humans. En ser un projecte de final de carrera hi ha hagut una sola persona treballant sota la direcció del gestor de projectes d'eyeOS i la supervisió del tutor a la universitat. No obstant, s'ofereix una simulació dels costos amb un equip real, que a eyeOS es compondria d'un analista, un dissenyador intern, un programador, un tester i el cap de projecte.

Recurs	Salari
Cap de projecte	40 €/h
Analista	40 €/h
Dissenyador intern	35 €/h
Programador	20 €/h
Tester	20 €/h

Taula 3: Recursos i salaris

D'acord amb l'assignació de recursos humans en cada tasca i la planificació temporal, es pot fer un càlcul del cost de desenvolupar el projecte. El calendari té dies de 5h.

²⁵ un *contrib* és una extensió de la biblioteca gràfica Qooxdoo que utilitza l'eyeOS. En aquest cas va ser necessari utilitzar el *contrib* SVG, de Marc Puts.

WBS	Nom	Comença	Acaba	Feina	Cost (€)
1	Fase d'inici	Mar 9	Mar 30	8d 1h	
1.1	Definició del producte	Mar 9	Mar 16	1d 7h	600
1.2	Formació	Mar 17	Mar 30	6d 2h	1,450
2	Fase de planificació	Mar 31	Abr 4	1d 7h	
2.1	Planificació	Mar 31	Abr 4	1d 7h	600
3	Fase d'execució	Abr 5	Jul 4	40d 5h	
3.1	Sistema base	Abr 5	Mai 9	15d 5h	
3.1.1	Especificació	Abr 5	Abr 6	1d 2h	400
3.1.2	Disseny	Abr 7	Abr 13	3d 1h	875
3.1.3	Implementació	Abr 14	Mai 5	10d	1,600
3.1.4	Proves	Mai 6	Mai 9	1d 2h	200
3.2	Sistema de gestió d'usuaris	Mai 10	Jun 6	12d 4h	
3.2.1	Especificació	Mai 10	Mai 16	3d 1h	1,000
3.2.2	Disseny	Mai 17	Mai 23	3d 1h	875
3.2.3	Implementació	Mai 24	Jun 2	5d	800
3.2.4	Proves	Jun 3	Jun 6	1d 2h	200
3.3	Sistema de compartició de pintades sobre les diapositives	Jun 7	Jul 4	12d 4h	
3.3.1	Especificació	Jun 7	Jun 13	3d 1h	1,000
3.3.2	Disseny	Jun 14	Jun 20	3d 1h	875
3.3.3	Implementació	Jun 21	Jun 30	5d	800
3.3.4	Proves	Jul 1	Jul 4	1d 2h	200
4	Fase de tancament	Jul 5	Ago 1	12d 4h	
4.1	Documentació	Jul 5	Ago 1	12d 4h	4,000
	Total			505h	15 480 €

Taula 4: Descomposició de tasques i costos

10.1.2 Recursos materials

Per desenvolupar el projecte ha fet falta

- Portàtil (640€) amb el següent programari lliure i gratuït:
 - Fedora 14 i entorn LAMP

- Netbeans, client de Subversion, eyeOS, Firefox 4+ i Google Chrome, iprelay, Firebug
- OpenOffice, Quick Sequence Diagram Editor, Dia, Inkscape, The Gimp i Gnome Planner
- Pantalla externa de 22" (200€)

També s'ha utilitzat la infraestructura de l'oficina de l'eyeOS. En concret, una màquina virtual per fer proves i el seu control de versions, però no es disposa de dades suficients per estimar-ne el cost.

10.1.3 Cost total

La suma de les hores de feina i els recursos materials,

$$Cost = 15480 + 640 + 200 = 16320 \text{ €}$$

11 CONCLUSIONS I TREBALL FUTUR

Tot i el resultat final satisfactori, encara queden alguns temes per millorar en el futur que s'han considerat fora de l'abast del projecte:

- *Usabilitat.* Alguns detalls d'usabilitat s'han deixat de banda per manca de temps i baixa prioritat. Per exemple, canviar el cursor en passar per sobre elements actius, fusionar les finestres de configuració i gestió de participants, aplicar un estil més acurat en les icones de la barra d'eines i fer que en esborrar un participant de la llista, torni a ser visible a la llista de contactes immediatament.
- *Seguretat.* L'eyeOS té algunes mancances de seguretat en els missatges de NetSync. Ha calgut crear un camp *from* en els missatges de l'eyeMeeting perquè, si bé el servidor n'escriu un, després no es pot consultar. És un error de disseny heretat de les primeres implementacions.
- *Funcions.* Mostrar totes les diapositives en miniatura, indicar sempre a l'usuari el seu tipus, poder triar eines de dibuix més complexes per fer les pintades...
- *Disseny intern:* definir encara millor el concepte de PresentationManager i crear-hi un sistema de memòria cau real. Utilitzar components VirtualList en comptes de Form.List de Qooxdoo i intentar utilitzar objectes Participant en comptes de matrius bàsiques. Potser en properes versions de Qooxdoo (1.3+) serà possible.

No obstant, aquest projecte ha estat satisfactori perquè desenvolupar l'eyeMeeting m'ha permès comprovar que he assolit les competències professionals relacionades amb aquesta tasca pròpies d'un enginyer tècnic en informàtica.

La conclusió més important i general és l'assoliment dels objectius proposats. D'altres són:

- *Emprenedoria i innovació.*
L'aplicació proposada a l'equip de l'eyeOS és única i nova, pretén solucionar un problema en el sistema i afegir valor al producte.

- *Sostenibilitat compromís social*

L'eyeOS i els seus components externs són programari lliure i tenen una comunitat al voltant. Escriure codi lliure contribueix a la democratització de l'accés al coneixement, al progrés de la informàtica i a la transparència. He pogut ser un membre actiu de la comunitat de Qooxdoo i de l'eyeOS, pels qui he emplenat informes d'errors que he detectat i solucionat i a qui he demanat ajuda i idees per solucionar problemes de manera creativa, polida i formal. Aquesta manera de treballar m'ha permès parlar directament amb el cap de desenvolupament de Qooxdoo i amb l'autor del contrib SVG, Marc Puts de la Universitat de Maastricht.

A més a més, tot el projecte i no només l'aplicació s'ha desenvolupat amb programari lliure i s'ha enregistrat en format de document obert.

- *Tercera (i quarta) llengua*

He pogut demostrar un altre cop que l'anglès pot ser llengua vehicular quan les circumstàncies ho requereixen. La comunicació amb altres desenvolupadors d'arreu del món, la documentació de Qooxdoo, el codi de l'eyeOS, o el codi i documentació que jo mateix he escrit han estat íntegrament en anglès. Durant el projecte (però fora del seu abast) he adquirit suficients coneixements tècnics per poder escriure en anglès pràcticament sencer el manual de desenvolupament que s'utilitzarà d'ara endavant per la comunitat, l'organització i els socis.

L'oficina d'eyeOS és particularment interessant amb els idiomes. S'hi parla majoritàriament català; també castellà, anglès, francès, grec, rus i xinès. També s'hi pot parlar alemany i anteriorment s'hi parlava italià. Si bé l'anglès és per mi una llengua estrangera molt interioritzada per l'Erasmus a Escòcia, el francès ho és a nivell bàsic i anecdòticament l'he hagut de fer servir.

- *Comunicació eficaç*

Amb l'escriptura de la memòria, la seva presentació, l'emplenament d'informes d'error, la creació de documentació tècnica o les discussions eficients i eficaces sobre l'abast del producte, disseny intern i altres aspectes, he pogut demostrar que puc comunicar-me de manera precisa, acurada i estètica amb un equip de persones presents o a través de la

xarxa en català i anglès com a mínim.

- *Treball en equip*

Si bé aquest ha estat el primer projecte gran desenvolupat per mi sol com s'espera que ho faci un enginyer tècnic, hi ha hagut treball en equip amb els dissenyadors, el meu tutor, el director del projecte, la comunitat de Qooxdoo, etc.

- *Aprenentatge autònom*

L'eyeMeeting ha estat creat a partir dels coneixements adquirits amb un manual de 30 pàgines sobre l'eyeOS, un llibre de JavaScript, documentació de Qooxdoo i molta lectura de codi ja fet. Ha estat crític saber distingir la informació correcta de la no útil i, com ha quedat demostrat, s'ha fet amb èxit.

- *Desenvolupament d'una solució software*

S'han aplicat els coneixements de totes les assignatures de la carrera relacionades amb el projecte, com DABD, DSBW, ES1 i 2, PROP, GSI, XCA... i tot i tractar-se d'una aplicació d'alt nivell i distribuïda, ha estat molt útil tenir coneixements de sistemes operatius (USO). Cal recordar que l'eyeOS imita el seu comportament.

El raonament crític, lògic i matemàtic han regit tot el procés perquè fos precís i sistemàtic. De la mateixa manera, la capacitat d'anàlisi i de síntesi han estat vitals per poder gestionar bé tota la informació que ha calgut tractar i poder trobar la direcció adequada al llarg de les setmanes.

Del projecte en queda la satisfacció de la feina ben feta en l'eyeOS, on m'he sentit part de l'equip format majoritàriament per estudiants o enginyers de la FIB de menys de 25 anys, i l'haver pogut donar una empenta a la iniciativa que el 2005 van tenir en Pau Garcia-Milà i en Marc Cercós. En queda la satisfacció d'haver demostrat en un entorn real els coneixements i la manera de fer adquirits a la facultat i una contribució al programari lliure.

12 BIBLIOGRAFIA I REFERÈNCIES

12.1 Obres consultades

- **[Conallen]** Conallen, J. *Building web applications with UML*. 2a ed. Addison Wesley, 2002.
- **[DABD]** Abelló, A; Rollón, E; Rodríguez, E. *Database design and administration*. Edicions UPC, 2008.
- **[Fowler]** Fowler, M. *Patterns of Enterprise Application Architecture*. Addison-Wesley, 2002. ISBN 978-0321127426.
- **[Cockburn]** Cockburn A. *Writing Effective Use Cases*. 1a ed: Addison-Wesley 2000, ISBN 0-201-70225-8.
- **[Pressman]** Pressman R. S. *Software Engineering: A Practitioner's Approach*. 6a ed: McGraw-Hill, 2005. ISBN: 0072853182
- **[Larman]** Larman, C. *Applying UML and Patterns*. Prentice Hall, 2005. ISBN 0-13-148906-2.
- **[Crockford]** Crockford, D. *JavaScript: The Good Parts*. 1a ed: O'Reilly, 2008. ISBN 978-0-596-51774-8
- **[Stefanov]** Stefanov. S. *JavaScript: Patterns*. 1a ed. O'Reilly, 2010. ISBN 978-0-596-80675-0
- **[Kappel]** Kappel [et al.]. *Web Engineering*. Wiley, 2006. ISBN 978-0470015544
- **[ES1]** Costal, D; Franch, X.; Ribera Sancho, M.; Teniente, E. *Enginyeria del software: especificació. Especificació de sistemes orientats a objectes amb la notació UML*. 3a ed: Edicions UPC, 2005. ISBN 84-8301-799-7
- **[DSBW]** Farré, C.; Rull, G. *Diapositives de disseny de sistemes basats en web*, 2009-2010, FIB – UPC.
- **[ES2]** *Diapositives de disseny d'enginyeria del software 2*, 2009, FIB –

UPC.

- **[ER]** Enginyeria de Requisits; FIB. *Requirements Specification Template* (Basada en *Volere Requirements Specification Template* d'Atlantic Systems Guild Limite, 2009). Ed. Febrer 2011.
- **[Emmerich]** Emmerich, W. *Engineering Distributed Objects*. London University - Wiley, 2000 ISBN: 978-0-471-98657-7

12.2 Recursos electrònics

- **[eyeosTeam]** The eyeOS Team. *Manual de desarrollo para eyeOS 2.x*. 1a ed: 2010. [Consulta: 9 de març 2011] Disponible a: http://eyeos.org/dev_manual.pdf
- **[Gamonal]** The eyeOS Team; Gamonal, E. *Developing for eyeOS 2.x*, 2a ed: Octubre 2011.
- **[Youseff]** Youseff, L.; Butrico, M.; Da Silva, D. *Toward a Unified Ontology of Cloud Computing*. En Grid Computing Environments Workshop, pàgines 1-10 (2008) [Consulta: maig 2009] Disponible a: <http://www.cs.ucsb.edu/~lyouseff/CCOntology/CloudOntology.pdf>
- **[Herrington]** Herrington, J. *Five common PHP design patterns*. 18 juliol 2006. Disponible a: <http://www.ibm.com/developerworks/library/os-php-designptrns/>
- **[MDN]** Mozilla Developers Network. Articles diversos. Disponible a: <https://developer.mozilla.org>
- **[SoftCatalà]** Softcatala. *Recull de termes*. Versió 4.2, 2010. Disponible a : <http://www.softcatala.org/recull.html>
- **[IEC]** Institut d'Estudis Catalans. *Diccionari de la llengua catalana*. 2a ed. Disponible a: <http://dlc.iec.cat/>
- **[Termcat]** Termcat. Terminologia i fraseologia dels productes informàtics. 2011. Disponible a: http://www.termcat.cat/ca/Diccionaris_En_Linia/123/
- **[QxDoc]** Qooxdoo team. *Documentation*. 2011. Disponible a

<http://qooxdoo.org/documentation>

- **[W3C-SVG]** W3C. *Scalable Vector Graphics (SVG) 1.1. 2a ed: 2011.*
Disponibile a <http://www.w3.org/TR/SVG/>